UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO VALLE DE CHALCO

IMPLEMENTACIÓN DEL MÉTODO DE ADMINISTRACIÓN DE LLAVES DUKPT EN EL SIMULADOR ASSET PARA TRANSACCIONES FINANCIERAS EN DISPOSITIVOS PUNTO DE VENTA

MEMORIA DE EXPERIENCIA LABORAL

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A

ERIK ALBA MARTINEZ

ASESORA:
DRA. EN M. E. MAGALLY MARTINEZ REYES

Revisora:
DRA. EN C. ED. ANABELEM SOBERANES MARTÍN

Revisor:
M. EN B.T. JUAN MANUEL SÁNCHEZ SOTO

VALLE DE CHALCO SOLIDARIDAD, MÉXICO OCTUBRE 2017.

Valle de Chalco, Méx. jueves, 24 de agosto de 2017

VOTO APROBATORIO DE ASESOR

M. EN P. J. JUAN CARLOS HERNÁNDEZ HERNÁNDEZ
SUBDIRECTOR ACADÉMICO
DEL CENTRO UNIVERSITARIO UAEM VALLE DE CHALCO

PRESENTE.

Por este conducto, comunico a usted que el trabajo de Memoria de experiencia laboral titulado:

Implementación del Método de Administración de Llaves DUKPT en el Simulador ASSET para Transacciones Financieras en Dispositivos Punto de Venta

Llevado a cabo por Erik Alba Martínez con número de cuenta 9923018 de la licenciatura en **Ingeniería en Computación** Registrado con fecha 14 de julio de 2017 y con número de registro ICO/02.08.17/476 ha concluido, por lo que **con fundamento en el artículo 93 del reglamento de Evaluación Profesional** *Que en letra dice "En el momento en que el asesor y el pasante consideren que el trabajo escrito está concluido, éste lo informará por escrito a la subdirección y solicitará la revisión del mismo por los profesores nombrados para tal efecto*".

Solicito la revisión del mismo por los profesores nombrados para tal efecto.

Sin más por el momento quedo de usted.

A T E N T A M E N T E

Dra. en M. E. MAGALLY MARTÍNEZ REYES
mmreyes@hotmail.com

OFICIO: FT5
Valle de Chalco, Méx. martes, 12 de septiembre de 2017

M. EN P. J. JUAN CARLOS HERNÁNDEZ HERNÁNDEZ
SUBDIRECTOR ACADÉMICO
DEL CENTRO UNIVERSITARIO *UAEM* VALLE DE CHALCO

PRESENTE.

Por este conducto, comunico a usted que el trabajo de Memoria de Experiencia Laboral con el título:

IMPLEMENTACIÓN DEL MÉTODO DE ADMINISTRACIÓN DE LLAVES DUKPT EN EL SIMULADOR ASSET PARA TRANSACCIONES FINANCIERAS EN DISPOSITIVOS PUNTO DE VENTA

Llevado a cabo por Erik Alba Martinez con número de cuenta 9923018 de la licenciatura en **Ingeniería en Computación** registrado el 14 de julio de 2017 con Número de Registro ICO/02.08.17/476 ha concluido y estamos de acuerdo para la impresión definitiva de Memoria de experiencia laboral

	Nombre	**Firma**
Asesor	DRA. EN M. E. MAGALLY MARTÍNEZ REYES	
Revisor	DRA. EN C. ED. ANABELEM SOBERANES MARTÍN	
Revisor	M. EN B.T. JUAN MANUEL SÁNCHEZ SOTO	

Sin más por el momento quedo de usted.

A T E N T A M E N T E

ERIK ALBA MARTÍNEZ

CARTA DE CESIÓN DE DERECHOS DE AUTOR

El que suscribe Erik Alba Martínez Autor(es) del trabajo escrito de evaluación profesional en la opción de Memoria de Experiencia Laboral con el título Implementación del Método de Administración de Llaves DUKPT en el Simulador ASSET para Transacciones Financieras de dispositivos Punto de Venta, por medio de la presente con fundamento en lo dispuesto en los artículos 5, 18, 24, 25, 27, 30, 32 y 148 de la Ley Federal de Derechos de Autor, así como los artículos 35 y 36 fracción II de la Ley de la Universidad Autónoma del Estado de México; manifiesto mi autoría y originalidad de la obra mencionada que se presentó en Ciudad de México para ser evaluada con el fin de obtener el Título Profesional de Ingeniero en Computación.

Así mismo expreso mi conformidad de ceder los derechos de reproducción, difusión y circulación de esta obra, en forma NO EXCLUSIVA, a la Universidad Autónoma del Estado de México; se podrá realizar a nivel nacional e internacional, de manera parcial o total a través de cualquier medio de información que sea susceptible para ello, en una o varias ocasiones, así como en cualquier soporte documental, todo ello siempre y cuando sus fines sean académicos, humanísticos, tecnológicos, históricos, artísticos, sociales, científicos u otra manifestación de la cultura.

Entendiendo que dicha cesión no genera obligación alguna para la Universidad Autónoma del Estado de México y que podrá o no ejercer los derechos cedidos.

Por lo que el autor da su consentimiento para la publicación de su trabajo escrito de evaluación profesional.

Se firma la presente en la ciudad de México, a los 02 días del mes de octubre de 2017.

ERIK ALBA MARTINEZ

Ciudad de México, a 26 de septiembre de 2017.

Universidad Autónoma del Estado de México,
Centro Universitario Valle de Chalco.
Avenida Hermenegildo Galeana número 3,
colonia Maria Isabel, municipio de Valle de
Chalco Solidaridad, C.P. 56615, Estado de
México.

Asunto: Autorización de uso de la Marca **PROSA** y **RED**.

A quien corresponda:

Gabriela García Ortiz, en mi carácter de apoderado legal de Promoción y Operación, S.A. de C.V., (en lo sucesivo "**PROSA**"), personalidad que acredito en términos de la escritura pública número 34,561 de fecha 16 de agosto de 2017, otorgada ante la fe del licenciado Heriberto Castillo Villanueva, titular de la notario número 69 del Distrito Federal (ahora Ciudad de México), misma que a la fecha no me ha sido modificada, limitada ni revocada en forma alguna, hago de su conocimiento lo siguiente:

Por medio de la presente, **PROSA** autoriza al señor **Erik Alba Martínez** el uso no exclusivo de la denominación social **"Promoción y Operación, Sociedad Anónima de Capital Variable"** o de su Abreviatura **"PROSA"**, así como de la marca **"RED",** propiedad de **PROSA**, para ser utilizadas, única y exclusivamente, dentro del trabajo final de Memoria de Experiencia Laboral para obtener el título de Ingeniero en Computación, el cuál llevará por nombre **"Implementación del Método de Administración de Llaves DUKPT en el Simulador ASSET para Transacciones Financieras en Dispositivos Punto de Venta"** y que en ningún caso podrá hacer un uso diferente al que por el presente se autoriza.

Así mismo, el señor **Erik Alba Martínez** no podrá ceder, ni conceder, ni sub-licenciar la denominación social de **Promoción y Operación, Sociedad Anónima de Capital Variable"** o de su Abreviatura **"PROSA"**, así como de la marca **"RED"** a terceras personas.

Sin más por el momento, envío un cordial saludo.

Atentamente,
Promoción y Operación,
S.A. de C.V.,

Lic. Gabriela García Ortiz
Apoderado Legal.

AGRADECIMIENTOS

A mi padre Lucio Alba, por el apoyo, el cariño, los consejos, por enseñarme a ver la vida como él la ve, por ser mi ejemplo a seguir y sobre todo por ser el mejor amigo que la vida me dio. Gracias Cuervo.

A mi madre Lourdes Martínez por su entrega incondicional, por ser la mejor confidente y por creer en mí muchas veces más que yo mismo. Gracias Amá, eres la mejor.

A mi pareja Rosalía Godínez por acompañarme en este viaje, a veces disfrutando y otras padeciendo, pero siempre junto a mí. Gracias Fleita.

A mis maestros de la escuela pero mejores maestros de la vida, Adolfo Sánchez y Víctor Sánchez, Gracias Perro, Gracias Fis.

A mis amigos y compañeros de profesión Guadalupe Ruiz y Juan Carlos Sánchez, por compartirme parte de sus valiosos conocimientos de este hermoso medio en el que trabajamos. Gracias Wangues, Gracias Juanito.

A mi líder Álvaro Ontiveros por la confianza brindada de aprender y aportar a la organización en los últimos 12 años y sobre todo por el impulso para concluir este proceso de mi formación académica. Gracias Álvaro.

A las Doctoras Magally Reyes y Anabelem Soberanes que me apoyaron e instruyeron durante la carrera y ahora un poco después en la elaboración de este trabajo. Mil Gracias profesoras.

A todas las personas que en algún momento dedicaron parte de su tiempo en cuidar de mi hija Erika mientras yo estudiaba y Rosy trabajaba, en especial a Doña Severiana, Daniela, Mary y a mi hermana Lucia Alba, siempre se los agradeceré.

Muchas Gracias a todos…

DEDICATORIA

A mi hija Erika Alba…

Mi primer gran motivo de hacer que las sucedan para bien, por creer en mí y porque si hubiera tenido que elegir ser mujer, habría elegido ser como ella.

A mi hijo Eduardo Alba…

El complemento perfecto de mi vida, mi gran inspiración, quien me recuerda todos los días que el amor existe.

Los amo y estoy orgulloso de ustedes.

IMPLEMENTACIÓN DEL MÉTODO DE ADMINISTRACIÓN DE LLAVES DUKPT EN EL SIMULADOR ASSET PARA TRANSACCIONES FINANCIERAS EN DISPOSITIVOS PUNTO DE VENTA

ÍNDICE

I. RESUMEN 10

II. IMPORTANCIA DE LA TEMÁTICA 12

III. DESCRIPCIÓN DEL PUESTO O EMPLEO 17

Título del Puesto 17
Objetivo del Puesto 17
Organigrama 17
Funciones y Responsabilidades del Puesto 18

IV. PROBLEMÁTICA IDENTIFICADA 24

V. INFORME DETALLADO DE LAS ACTIVIDADES 26

Protocolos de Mensajería 26
Datos a ser cifrados 26
Algoritmo de encriptación Triple DES 26
Generación de Llaves aleatorias 27
Implementación del algoritmo RSA para traslado de llaves 27
Integración de componentes DUKPT en ASSET 27
Variante de comandos Thales en HSM 28
Inicialización y carga de llaves 28
Operación de Scripts ASSET Adquirentes HPDH e ISO8583 28
Pruebas de flujo transaccional Emisor-Adquirente con DUKPT 28
Interpretación de códigos de rechazo en ASSET 29

VI. SOLUCIÓN DESARROLLADA Y SUS ALCANCES 30

¿Qué es la información sensible? 30
Criptografía 30
Criptografía Simétrica 30
Criptografía Asimétrica 31
Normativas 32

Hardware de Seguridad HSM's Thales 33
Manejo de Llaves de Encripción 34
Descripción de Datos Sensitivos 35
Track 2 35
Código de Seguridad (CVV2 / CVC2) 36
Track1 37
Solución desarrollada 38
Almacenar e informar los datos actuales de la Terminal Punto de Venta 38
Solicitar ó leer Datos Sensitivos dependiendo del modo de ingreso y cifrarlos usando 3DES con llave Simétrica DUKPT "Actual" 41
Implementar DUKPT como método de Administración de Llaves para el cifrado de Datos Sensitivos. 48
Algoritmo de DUKPT en el Script Adquirente de ASSET 50
Ejemplos de datos cifrados bajo el método de DUKPT en ASSET 52
Mantener e Informar el Contador Real de Cifrados 54
Carga Remota de Llaves usando cifrado RSA con Llaves Públicas 55
Ejemplo de Carga Remota de Llaves en Script Adquirente de ASSET 58
Validación de Firma de Llaves RSA Públicas 62
Cargar de tabla de BINES para determinar si se deben cifrar los Datos dependiendo del prefijo de la tarjeta 63
Operar sin cifrado de datos mientras no haya inicialización de llaves 64
Calcular / Validar el CRC32 para campos cifrados 65
Pruebas de simulación de datos sensibles cifrados en AASET 68
Solicitud de Inicialización de Llaves 68
Transaccionando con Cifrado posterior a la carga de llaves 75
Transaccionando sin Cifrado 80

VII. IMPACTO DE LA EXPERIENCIA LABORAL 83

VIII. REFERENCIAS DE CONSULTA 85

IX. ANEXOS 88

I. RESUMEN

La seguridad de la información dentro de los Sistemas Bancarios es un tema crítico a tener en cuenta cuando se trata de procesamiento de transacciones electrónicas, el actuar y aplicar de forma alineada a los estándares y normas nacionales e internacionales establecidos, es el compromiso que deben asumir responsablemente las instituciones así como los expertos en Tecnologías de la Información participantes en mencionado proceso.

En este trabajo se abordan los puntos más importantes relacionados con la seguridad del proceso de transacciones en línea con tarjetas de crédito y débito realizadas en puntos de venta de comercios minoristas (Retail), así como los utilizados en cadenas comerciales (Interredes) y cámaras de compensación que operan como Switches Transaccionales Bancarios dentro la república mexicana.

Se hace énfasis en el proceso de cómo se garantiza la seguridad de la información más importante de los usuarios de tarjetas de crédito y débito que se encuentra contenida en los plásticos que las diferentes instituciones bancarias ofrecen a sus clientes como instrumento para tener acceso a sus cuentas de fondos monetarios, créditos otorgados o a los productos que estas ofrezcan a fin de eliminar el uso de dinero en efectivo cuando se realice la adquisición de algún servicio o producto.

El análisis principal se basa en cómo se establecen los mecanismos utilizando tecnologías de simulación en las que puede implementarse el método de administración de llaves DUKPT como el desarrollado en los puntos de venta electrónicos y los procesadores bancarios de transacciones.

Al tener la responsabilidad de operar un sistema como el de un banco o un procesador de transacciones, se deben tomar continuamente decisiones acerca de las acciones que este debe ejecutar. Estas decisiones deben ser tales que el comportamiento que se entregue como resultado satisfaga de la mejor manera posible los objetivos planteados en los procesos de certificación.

Para poder decidir correctamente es necesario saber cómo responderán los sistemas ante una determinada acción. Esto podría hacerse experimentando con los sistemas productivos; pero factores como costos, seguridad y otros, hacen que esta opción sea generalmente no viable. A fin de superar estos inconvenientes, se reemplaza al sistema real por otro que en nuestro caso es una versión simplificada del productivo y mediante el simulador de transacciones ASSET (por sus siglas en inglés, *ACI Simulation Services for Enterprise Testing*) reproducimos el comportamiento de los dispositivos electrónicos Puntos de Venta. Realizado lo anterior obtenemos el modelo a utilizar para llevar a cabo las experiencias necesarias sin los inconvenientes mencionados.

Los resultados obtenidos serán utilizados para continuar con los demás procesos de certificación implicados con el flujo transaccional de acuerdo a la solicitud del cliente o proyecto que se esté atendiendo en su momento.

II. IMPORTANCIA DE LA TEMÁTICA

Este trabajo es desarrollado en la empresa Promoción y Operación SA de CV (PROSA), fundada en el año de 1968 por un consorcio de cinco bancos con el fin de concentrar el procesamiento transaccional entre ellos, actualmente es el "Switch Transaccional Bancario" encargado de recibir, procesar y enviar operaciones bancarias que involucran mayormente tarjetas de débito y crédito, y algunos otros productos o servicios complementarios, esto lo convierte en líder a nivel América Latina, gracias a la aceptación medios de pago en México y en el resto del mundo.

Los principales servicios que ofrece son: ruteo, autorización, compensación y liquidación[1] de transacciones electrónicas, así como la operación lógica de Terminales Punto de Venta (TPV's) también llamadas POS, por sus siglas en inglés, *Point Of Sale*, Cajeros Automáticos (ATM, por sus siglas en inglés, *Automatic Teller Machine*), aplicaciones de Internet relacionadas con medios de pago y otros servicios afines de valor agregado para sus clientes.

Entre los clientes más importantes de la organización se encuentran instituciones bancarias, cajas de ahorros, interredes de comercios Retail, compañías telefónicas y diferentes cámaras concentradoras de transacciones bancarias como las empresas Eglobal, Total System, American Express, VISA y Mastercard, entre otras.

Basado en el nivel de exigencia por garantizar la seguridad de la información de los tarjetahabientes contenida en sus tarjetas utilizadas como medio de pago, se presenta la necesidad de incorporar dentro del procesamiento transaccional, diferentes mecanismos de seguridad que estén a

[1] Proceso por el cual se cobra el dinero por lo adquirido a la cuenta del nuevo propietario y, de forma simultánea, se abona al antiguo propietario el dinero correspondiente a la venta de sus artículos o por la disposición del efectivo que se haya realizado (Introducción a las Finanzas, 2016).

la vanguardia y que cumplan con los estándares internacionales que dictan los organismos encargados de regular y preservar la integridad de los datos implicados en transacciones electrónicas de tipo financieras.

La práctica empleada más común para proteger la información contenida en la banda magnética y los chips de las tarjetas bancarias es el uso de la criptografía, la cual se aplica cifrando la información clave del cliente como son: Numero Personal de Identificación (NIP) que puede ser de cuatro hasta 12 dígitos, Track2 que se refiere a los números impresos en la tarjeta que van de los 15 a los 19 dígitos, fecha de expiración de la tarjeta y código de verificación CVV2 (para tarjetas VISA) ó CVC2 (en tarjetas Mastercard) que en su mayoría es de tres posiciones y se encuentra al reverso del plástico, entre otros datos.

En la actualidad uno de los algoritmos más utilizados para el cifrado de datos en transacciones electrónicas es el "Triple DES", desarrollado por IBM en el año de 1998, y se encarga de aplicar tres veces el algoritmo DES que se basa en el manejo de llaves de 56 bits, dando por resultado al ejecutarlo tres veces una longitud de clave utilizada de 168 bits. Aunque su efectividad continúe siendo de 112 bits por el proceso que realiza de encriptación[2] de datos, hasta hoy no se tiene registro de ataques de fuerza bruta que hayan sido capaces de vulnerar la información cifrada bajo este.

A pesar de que existen otros algoritmos de encriptación más eficaces y rápidos que el Triple DES, como el Estándar de Encriptación Avanzada (AES, por sus siglas en inglés, *Advanced Encryption Standard*)[3], Triple DES se mantiene como estándar para el cifrado de datos en transacciones con tarjetas de crédito y débito por la alta efectividad y niveles de seguridad aun cuando al triplicar el número de operaciones lo vuelva más lento (DEVNULL, 2016).

[2] Proceso para enmascarar una cierta información de carácter confidencial. (Amparo Fuster Sabater, 1998).
[3] AES es una técnica de cifrado de clave simétrica. El algoritmo AES utiliza una de las tres fortalezas de clave de cifrado: una clave de encriptación (contraseña) de 128-, 192-, o 256- bits. (Bitzipper, 2017).

Sin embargo, debido a que la información se tiene que trasladar de un lugar a otro se debe tener especial atención al uso y tiempo de vida de las llaves, ya que dicho algoritmo al estar basado en el uso de llaves simétricas (es decir, la clave secreta la deben tener los dos usuarios), si alguna es comprometida en el punto de envió o recepción, la seguridad de la información quedaría expuesta para todas las transacciones que utilicen dichas llaves.

A fin de incrementar el nivel de seguridad que provee el algoritmo de encriptación Triple DES, la Comisión Nacional Bancaria (CNVB) en conjunto con la Asociación de Bancos de México (ABM) acordaron implementar el Método de Administración de Llaves DUKPT (por sus siglas en inglés *Derived Unique Key per Transaction*), el cual consiste en generar una llave aleatoria de 128 bits Triple DES, derivada de una llave Inicial bajo una serie de parámetros generados por terminal, para lograr obtener una llave Triple DES por cada transacción que se realice en el dispositivo de punto de venta y que del lado del switch o autorizador esta pueda ser regenerada para descifrar el dato y así continuar con el proceso de autorización de la transacción hacia el banco emisor. Con esto lo que se busca es que en el caso de quedar comprometida la llave origen o destino el riesgo sería minimizado a una sola transacción.

Dicha implementación impacta a diferentes áreas tanto de la compañía como a los fabricantes y desarrolladores de software de terminales punto de venta, como también a los diferentes bancos y cadenas comerciales que cuentan con aceptación de pagos con tarjeta. Por una parte, fue necesario para los bancos que ofrecen el servicio de Retail con dispositivos punto de venta evaluar las características de sus terminales en producción y con esto determinar la inversión que implicaría el cambiar a dispositivos con soporte para el manejo de Método DUKPT, y en el caso de los dispositivos que tuvieran soporte se evaluaría únicamente el impacto en desarrollo y reprogramación de estas.

Por su parte, las diferentes áreas de desarrollo de software de los proveedores tendrían que delimitar el impacto en tiempo y costo que conllevaría aplicar la nueva normativa de la implantación de DUKPT para satisfacer la demanda de actualizaciones que tendrían por parte de los bancos dueños de terminales punto de venta.

En el caso de cadenas comerciales como Walmart, Soriana, Comercial Mexicana, Chedraui, entre otras, fue necesario evaluar los cambios desde sus cajas receptoras de pagos, su host de procesamiento central y hasta la conexión con los switches de PROSA o Eglobal según correspondiera a su ruteo de transacciones.

PROSA por ser el switch con mayor número de transacciones y con más clientes activos, fue designado como responsable de generar y resguardar la primera llave utilizada por el Método DUKPT llamada también Llave Base que servirá como maestra para todas las llaves iniciales que serán cargadas en los dispositivos punto de venta. Con esto PROSA tendría la responsabilidad de realizar el mayor número de desarrollos en sus sistemas, así como efectuar el proceso de certificación correspondiente para cada una de las entidades y dispositivos implicados en el proceso transaccional.

Dentro de los procesos de certificación para asegurar el cumplimiento de las nuevas funcionalidades se establecieron diferentes etapas que irían desde la validación de cada uno de los modelos de terminales con sus respectivas versiones de software, hasta la generación de llaves y desencriptación de los datos por parte del aplicativo central del switch de PROSA.

Debido a que tanto los dispositivos punto de venta como la aplicación central de procesamiento llamada BASE24 se encontraban en desarrollo, el uso de la simulación se vuelve vital, ya que aquí se establecen los niveles de cumplimento esperados en todo el proceso de aceptación.

Actualmente, la empresa cuenta con varias herramientas de simulación, pero nuestro caso de estudio se concentró en el simulador ASSET desarrollado por la empresa “ACI Worlwide Ltd” en Omaha Nebraska en el año 2006, y que es capaz de realizar la simulación de mensajes electrónicos mediante el uso de archivos de órdenes de procesamiento por lotes que usualmente son simples y almacenados como texto plano (SCRIPTS), que el usuario puede programar en un leguaje propietario, así como definir diferentes tipos de mensajes estándar utilizados en el procesamiento transaccional.

Algunos de los formatos de transacciones bancarias estándar desarrollados en este simulador son el ISO8583 en sus versiones 87 y 93 utilizado por PROSA y los de las diferentes marcas globales de procesamiento electrónico como VISA, Mastercard, Total System, First Data, American Express, así como también el formato propietario de los Dispositivos Punto de Venta llamado HPDH (por sus siglas en inglés *Hypercom POS Device Handler*).

La importancia de implementar el Método de Administración de Llaves DUKPT dentro del simulador de los formatos utilizados por las terminales, es para contrarrestar la dependencia de tener que usar terminales punto de venta certificadas con DUKPT en cualquier certificación, ya que el desarrollo de su software también se encontraría en la fase se aceptación para su uso.

Las consideraciones para la implementación del Método pasan a ser exactamente las mismos que las definidas a los proveedores de TPV’s y el resultado esperado es contar con la facultad de simular una transacción cifrada y con la administración de los datos sensibles a la vez.

III. DESCRIPCIÓN DEL PUESTO O EMPLEO

Título del Puesto

Arquitectura de Aplicaciones

Objetivo del Puesto

Establecer y definir la arquitectura o modificaciones a la misma, de los sistemas y procesos operativos PROSA, derivados de la entrada de un nuevo servicio u operativa, utilizando metodologías de arquitectura de negocio y tecnología.

Organigrama

Ubicación estructural del área de Arquitectura de Aplicaciones en PROSA, ver imagen 1.

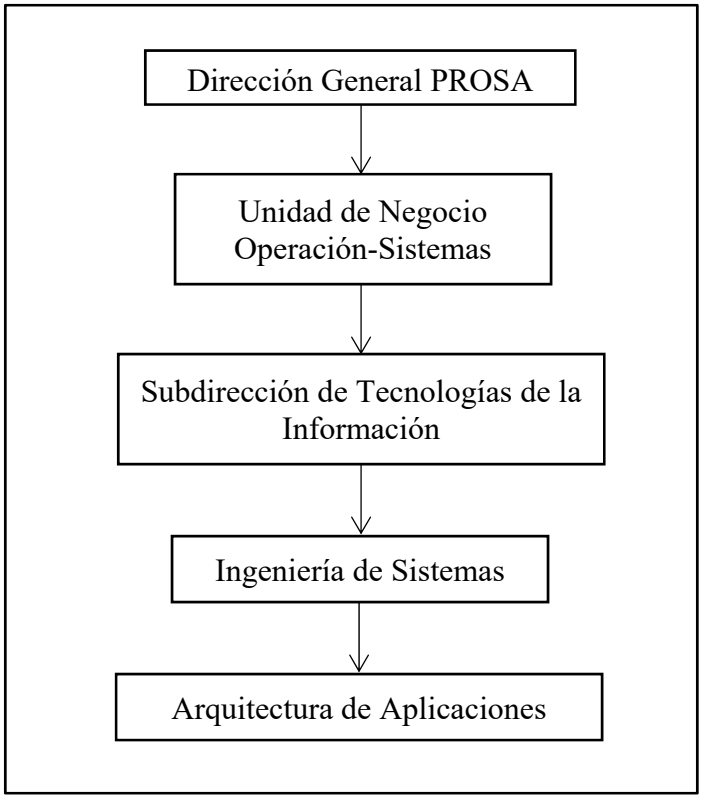

Imagen 1. Estructura Organizacional Sistemas Operación. (PROSA, 2016).

Funciones y Responsabilidades del Puesto

A continuación se enlistan las principales funciones y responsabilidades a desempeñas por parte del candidato a desempeñar el puesto de Arquitecto de Aplicaciones.

I. Definir la arquitectura tecnológica y de procesos, nuevos y existentes, de los servicios PROSA, asegurando el cumplimiento de los Acuerdos de Niveles de Servicio (SLA's, por sus siglas en inglés, *Service Level Agreement*)[4], Acuerdos de Nivel de Operación (OLAs, por sus siglas en inglés, *Operation Level Agreement*)[5], criterios de aceptación, viabilidad técnica, comercial y económica, con base al análisis de diferentes tecnologías y esquemas de arquitectura; enseguida se enlistan las principales:

- Definir y documentar la arquitectura tecnológica y de procesos relacionada con cada nuevo requerimiento de los clientes.
- Generar estándares de arquitectura para la implementación o modificación de servicios PROSA.
- Actualizar los estándares de arquitectura para asegurar su uso y validez en PROSA
- Asesoría a clientes para la elaboración de propuestas de viabilidad de nuevos productos/servicios, considerando insumos tecnológicos, definiciones de procesos operativos, y uso de recursos, analizando productos del mercado y haciendo recomendaciones para la implementación del nuevo servicio.

[4] El objetivo principal de un Acuerdo de Nivel de Servicio es detallar absolutamente todos los elementos del Acuerdo de cara al cliente e internamente. (Information Technology Infrastructure Library, 2017).

[5] El objetivo último de este Acuerdo es disponer de una referencia sobre cómo proceder en el desarrollo del Servicio, con dos componentes fundamentales: es un documento completamente técnico, y es exclusivamente de uso interno en la Organización. (Information Technology Infrastructure Library, 2017).

- Generación de vistas de solución que integren las relaciones entre procesos, sistemas y actores involucrados, utilizando metodologías de modelado de sistemas.

- Evaluación de tecnologías o herramientas que permitan integrar la que mejor se adapte al cumplimiento de los requerimientos de los nuevos productos y/o servicios propuestos por los clientes.

- Desarrollar las diferentes arquitecturas de transición para las soluciones propuestas, que se alineen al ciclo de vida definido para el producto, contemplando la ampliación de capacidad tecnológica y de recursos, automatización y mejor infraestructura.

- Asesorar a áreas internas para la correcta implementación de la arquitectura tecnológica, durante la ejecución de cada proyecto.

II. Evaluar y Generar propuestas de mejora a la arquitectura tecnológica y de procesos de los Sistemas Core Bancario[6], para garantizar su flexibilidad, disponibilidad y buen desempeño, como son:

- Evaluar y proponer mejoras a procesos operativos y sistemas actuales con la finalidad de elevar sus niveles de disponibilidad, asegurar su integración a nuevas tecnologías y facilitar la adopción de estándares de terceros.

- Identificar inconsistencias en la arquitectura de los sistemas Core Bancario durante la definición de una nueva arquitectura, notificando al responsable la posible solución.

- Desarrollar la arquitectura para las Re-ingenierías a las aplicaciones Core Bancario.

- Recomendar mejoras al cálculo de capacity de aplicaciones Core Bancario.

[6] El Core Bancario se puede definir como un sistema de cómputo que integra mejores prácticas, reglas de negocio y procesos de banca para la operación de una institución financiera. (Gesfor México, 2017).

- Identificar en la generación de cada arquitectura, los riesgos e impactos que pudieran afectar a la planificación, tiempo y costo del proyecto.
- Identificar riesgos potenciales de recursos técnicos como de personal que puedan generarse con la integración de un nuevo servicio o proceso.
- Identificar problemas de diseño, implementación, interfaz, verificación y mantenimiento que pongan en riesgo el proyecto.
- Incluir en las propuestas de viabilidad los riesgos tecnológicos.
- Definir un plan de acción para el tratamiento de los riesgos identificados que puedan ocurrir en el proyecto.
- Implementar y mantener en operación los controles de seguridad de la información.

III. Realizar propuestas de viabilidad y de solución a los requerimientos de negocio, que contemplen la implementación y operación de nuevos procesos operativos, selección de software, bases de datos, librerías, frameworks, estándares tecnológicos; así como selección de infraestructura, sistemas operativos y sistemas de recuperación, dentro las cuales se detallan:

- Identificar y registrar los requerimientos de usuario sobre el nuevo proceso o servicio que desea implementar.
- Asesoría a los usuarios que generarán un nuevo producto o servicio para alinearlos con la tecnología, procesos y requerimientos técnicos y operativos existentes.
- Generar y documentar propuestas de viabilidad considerando tiempos y costos estimados de implementación del nuevo servicio o proceso.
- Acompañamiento con el cliente para la presentación de la propuesta de viabilidad.

IV. Proporcionar coaching y mentoring a diseñadores, para la generación de soluciones a los requerimientos realizador por los clientes de acuerdo con las siguientes actividades:

- Transmisión del conocimiento de los proyectos a diferentes niveles, a los mismos arquitectos, a comerciales, a diseñadores, certificadores y desarrolladores.
- Impartición de capacitaciones, ya sea de algún proyecto en específico o de los procesos de arquitectura de sistemas.
- Generación de estándares que permitan una mayor eficacia en la impartición de capacitaciones y documentar cada una de ellas.
- Asesorar a los diseñadores en el proceso de diseño de la solución de los requerimientos a los que sea asignado el arquitecto como coach.
- Gestión de estándares de arquitectura a aplicaciones.
- Supervisar al área de diseño para asegurar que los diseños de nuevos proyectos cumplan con la arquitectura de aplicaciones.
- Asegurar que la documentación generada para proveedores sobre proyectos normativos y necesidades internas cumple con los lineamientos especificados por arquitectura.
- Evaluar la lógica de desarrollo de proveedores que garantice un núcleo simple y cumplimiento con la arquitectura en los sistemas PROSA.
- Trabajar en conjunto con diseño para la atención a posibles fallas en producción por definiciones de arquitectura.
- Capaz de integrar conceptos de seguridad de la información en conjunto con dicha área.

V. Evaluar herramientas tecnológicas en el mercado cuando nuevos productos/servicios propuestos en PROSA y autorizados en un proyecto lo requieran, tales como:

- Investigación de tendencias tecnológicas, con el fin de identificar aquellas que puedan ser utilizadas en las arquitecturas generadas para cumplir más eficientemente los requerimientos de los clientes.
- Asesorar a los clientes sobre las nuevas tecnologías que pueden ser utilizadas para mejorar sus procesos.
- Proponer proyectos relacionados con la implementación de mejoras en procesos y tecnología basadas en nuevas herramientas tecnológicas.
- Difusión a diseño de nuevas herramientas adoptadas en PROSA.

VI. Registrar y mantener el mapa arquitectónico PROSA, a nivel servicios, procesos, aplicaciones, infraestructura y recursos, como son:

- Definir soluciones arquitectónicas que consideren las vistas de líneas de negocio, procesos, aplicaciones, datos, infraestructura, recursos y riesgos.
- Registrar las diferentes vistas de cada solución arquitectónica en el sistema de arquitectura empresarial.
- Mantener y utilizar los estándares definidos para la diagramación y registro de la arquitectura empresarial.
- Validar y autorizar modificaciones al ambiente productivo que contiene las definiciones de la arquitectura empresarial.
- Difundir y promover el uso de la herramienta de arquitectura empresarial.

Para garantizar el correcto desempeño de las funciones del puesto es importante que el candidato cuente con conocimientos con sobre alguna de las Metodologías de Arquitectura o Frameworks utilizados en PROSA, como The Open Group Architecture Framework (TOGAF) o alguno relacionado a este.

IV. PROBLEMÁTICA IDENTIFICADA

Como parte de la publicación del Cápitulo X “Seguridad de la Información” en 2009, la Comisión Nacional Bancaria y de Valores (CNVB) estableció que la información sensitiva contenida en las tarjetas de crédito y débito como son: número de tarjeta, fecha de expiración y código de verificación en otros daos, deberá ser protegida mediante el uso de medios electrónicos.

En cumplimiento con esta normativa se definió que dicha información sería protegida bajo cualquiera de los dos procesos siguientes: el primero que la información de la cuenta sea transportada mediante un canal seguro como el uso de certificados SSL y/o indicadores de comercio electrónico 3DSecure[7], en el segundo caso en donde no es posible contar el método anterior de seguridad, se definió que la información relacionada a la cuenta sea cifrada utilizando el algoritmo de encriptación triple DES bajo el Método de Administración de Llaves DUKPT.

A fin de cumplir con la normativa de protección de datos referente al uso del método DUKPT en los procesos de desarrollo y certificación de PROSA, se detectó la necesidad implementar dicha operativa en el simulador de transacciones ASSET para así contar con una herramienta capaz de simular el procesamiento realizado por los diferentes dispositivos de cobro con tarjetas y con esto validar el flujo transaccional completo sin la dependencia de contar con dispositivos físicos como Terminales Punto de Venta ó Pin Pads[8].

Para lograr la implementación del Método DUKPT fue necesario analizar la estructura y funcionamiento de todos los procesos que lo componen, así

[7] 3D-Secure es el protocolo que ofrece a los consumidores una mayor seguridad y una sólida autenticación cuando paguen con tarjeta de débito o crédito en Internet. Se denomina, en función del tipo de tarjeta, “MasterCard SecureCode”, “Verified by Visa” y, en el caso de las tarjetas American Express, “Safekey”. (Paypal, 2017)

[8] Un PinPad o equipo con entrada de pin es un dispositivo electrónico usado en transacciones con tarjetas de crédito o débito, que acepta y codifica el número de encriptación personal de un tarjetahabiente, para corroborar su identidad y permitir que se realice una venta, sin el uso de dinero físico. (Tecnocomputación 3000, 2017)

como identificar la manera en que estos serían desarrollados dentro del simulador.

Por su parte se sugirió realizar un bosquejo de las funciones con las cuales el simulador cuenta y que se realizan en el algoritmo de encriptación 3Des, de igual forma se evaluaron los requisitos del método de administración de llaves DUKPT de acuerdo con las definiciones emitidas por la CNBV y los Switches Bancarios para determinar el nivel de cumplimiento y en su caso el impacto a nivel de desarrollos dentro del simulador.

V. INFORME DETALLADO DE LAS ACTIVIDADES

De acuerdo con las actividades del proceso de implementación del Método DUKPT como parte del simulador ASSET se identificaron las siguientes:

Protocolos de Mensajería

Derivado del inventario de scripts que actualmente se encuentran desarrollados en el simulador ASSET, se determinaron los protocolos que serían parte de proceso de implementación a DUKPT, los cuales fueron: HPDH, utilizado como formato de mensajería para terminales conectadas directo a PROSA bajo el esquema Front End. El segundo formato a impactar sería el ISO8583 sobre el cual las cadenas comerciales procesan las transacciones enviadas a PROSA adquiridas en sus puntos de venta.

Datos a ser cifrados

Derivado de las disposiciones legales emitidas por la CNBV la información a ser cifrada será la siguiente:

a. El Track 2, que puede ser:
    i. El Track2 completo leído de banda o el equivalente contenido en el CHIP.
    ii. un Track2 “Parcial” construido en base al PAN + Fecha de Expiración, cuando estos datos son digitados.
b. El código de Seguridad (CVV2/CVC2/4DBC), cuando sea solicitado por el adquirente.
c. El Track 1 completo (sólo cuando hay lectura de banda).

Algoritmo de encriptación Triple DES

Se analizaron las funciones del simulador para la ejecución del algoritmo Triple DES dentro del simulador, el cual se aplicaría para cifrar la información utilizando como llave las subsecuentes derivadas del método DUKPT.

Generación de Llaves aleatorias

Se definió el mecanismo interno que el simulador realizaría para la generación de la llave inicial, la cual debería estar formada por 32 caracteres en hexadecimal y sería la base de la llave inicial que PROSA generaría para un dispositivo en particular bajo DUKPT.

Implementación del algoritmo RSA para traslado de llaves

Como se realiza físicamente en las terminales punto de venta y Pin Pads, se determinó el procedimiento que realizaría el simulador para cargar la llave pública RSA en el script adquirente, sería mediante el archivo de llaves dependiendo del formato de mensajería para el dispositivo que se estuviera emulando, ya sea ISO8583 ó HPDH.

Integración de componentes DUKPT en ASSET

Alineado a la estructura del Método de Administración de Llave DUKPT, se definió la manera en que el simulador realizaría el armado de cada uno de los elementos que lo componen, como son:

1.- Estructura de KSN

2.- Tokens de solicitud de información

2.1 Token ES. Estado del terminal y su configuración de cifrado, ve definición en anexo 2.

2.2 Token EW con los datos del PIN PAD para solicitar una nueva llave, ve definición en anexo 2.

3.- Tokens de respuesta

3.1 Token ER. Resultado del cifrado , ve definición en anexo 2.

3.2 Token EX. Datos de la nueva llave inicial a cargar en la TPV o PIN PAD, ve definición en anexo 2.

La definición completa de Tokens utilizados se encuentra en el Anexo 2.

Variante de comandos Thales en HSM

Conforme al estándar internacional de uso de DUKPT, se determinó la manera en que sería aplicada la variante a la llave inicial, la cual fue creada para uso específico de cifrado de información sensitiva en terminales punto de venta.

Inicialización y carga de llaves

Una vez realizada la integración de los módulos y funciones correspondientes al manejo de DUKPT como parte de los scripts adquirentes dentro del simulador, se dio paso a las pruebas de inicialización de llaves bajo los esquemas Front End (Terminales conectadas a PROSA, formato HPDH) y Back End (Terminales o Pin Pads conectados a un Host Externo, formato ISO8583).

Operación de Scripts ASSET Adquirentes HPDH e ISO8583

Ya con el proceso concluido de carga de llave inicial en el script adquirente como parte del simulador, se iniciaron las primeras pruebas en formato HPDH e ISO8583, en donde se validaría por una parte la correcta encriptación de los datos sensitivos, así como la aplicación del Método DUKP para derivación de llaves en transacciones futuras.

Pruebas de flujo transaccional Emisor-Adquirente con DUKPT

Se realizó el flujo completo operando de manera simultánea scripts simulando Terminales Punto de Venta “Front End-HPDH”, Pin Pads “Back End-ISO8583” donde se validó la encriptación de datos por parte del simulador Adquirente, la desencriptación de datos realizada por el Módulo de seguridad Thales conectado al Switch de PROSA “BASE24”, hasta su autorización por el Banco emisor de la tarjeta en cuestión, ver Anexo 1.

Interpretación de códigos de rechazo en ASSET

Para los casos en que se produce algún problema al momento de desencriptar la información sensitiva se definieron algunos códigos de respuesta sobre los cuales los dispositivos tomarían alguna acción en caso de presentarse.

En esta fase se validó el comportamiento y las acciones que tomarían los scripts adquirentes dentro del simulador basado en su definición correspondiente. Los códigos de rechazo a aplicarse se describen en el Anexo 3.

Como validación de las repuestas, en los scripts adquirentes se desarrollaron funciones específicas para interpretar dichos códigos, a su vez estas serán encargadas de notificar a través de la interface gráfica del simulador su presencia e información del mensaje para determinar su origen.

VI. SOLUCIÓN DESARROLLADA Y SUS ALCANCES

Actualmente, uno de los activos más importantes para las empresas de Tecnologías de la Información “TI”, es la información y especialmente la información sensible.

¿Qué es la información sensible?

“Es aquella información que, definida por su propietario, cuya revelación, alteración, pérdida o destrucción puede producir daños importantes a la organización” (Apoyo Informática, 2017).

Para nuestro caso de estudio podríamos decir que la información sensible: son los datos que están relacionados con las terminales punto de venta y las tarjetas de crédito/debito.

Criptografía

La palabra criptografía proviene del griego “criptos” que significa “oculto” y “grafe” escritura que alude textualmente a la “escritura oculta”. Es la ciencia utilizada para cifrar y descifrar información utilizando técnicas matemáticas que hagan posible el intercambio de mensajes de manera que solo puedan ser leídos por las personas a quienes van dirigidos (Seguridad Informatica Lean Lean, 2017).

Criptografía Simétrica

Es un algoritmo criptográfico en donde se utiliza una sola llave para cifrar y descifrar la información, por ejemplo: DES, 3DES y AES, ver Imagen 2.

En la criptografía simétrica solo utiliza una clave para cifrar y descifrar el mensaje que deben conocer el emisor y el receptor previamente, esto se convierte en un punto débil para los sistemas ya que en la comunicación de las

claves entre ambos sujetos, resulta más fácil interceptar una en cualquiera de los sitios que se encuentre almacenada y con esto lograr descifrar el mensaje en cualquiera de los puntos en que sea interceptado.

Imagen 2. Esquema de tipo de criptografia Simetríco. (Instinto Lógico, 2017).

Criptografía Asimétrica

Está basada en el uso de dos claves, una por cada entidad; es decir, una clave pública (se utiliza para cifrar) y una privada (se utiliza para descifrar). Como ejemplo se tiene el algoritmo RSA. Los métodos criptográficos garantizan que ese par de claves sólo se puede generar una vez, de modo que se puede asumir que no es posible que dos personas hayan obtenido casualmente el mismo par de claves, ver Imagen 3.

Imagen 3. Esquema de tipo de criptografia Asimetríco. (Instinto Lógico, 2017)

Normativas

En el Capítulo X del uso del servicio de Banca Electrónica de la CNBV Artículo 316 Bis 10 se estableció: Cifrar los mensajes o utilizar medios de comunicación cifrada, en la transmisión de la Información sensible del usuario procesada a través de medios electrónicos, desde el dispositivo de acceso hasta la recepción para su ejecución por parte de las Instituciones.

Por su parte el Estándar internacional PCI DSS Payment Card Industry – Data Security Standard establece: Cifrar la transmisión de los datos del titular de la tarjeta en las redes públicas; es decir, la información confidencial se debe cifrar durante su transmisión.

PROSA a través de estas estrategias y normativas emitidas por entidades nacionales e internacionales, ha logrado mitigar el fraude a través del método DUKPT. Esto permite que los tarjetahabientes hagan sus compras en terminales punto de venta de manera confiable y segura en cualquier momento y lugar dentro de la república mexicana, ver Imagen 4.

- En Terminales Punto de Venta Back-End, es responsabilidad de cada Host Bancario implementar la solución de cifrado de datos.

- Para las Terminales Punto de Venta Front-End, es responsabilidad de los Adquirentes implementar la solución de cifrado de datos.

- Respecto a la comunicación con los Bancos Emisores, es responsabilidad de ellos junto con PROSA, implementar una solución de cifrado de canal.

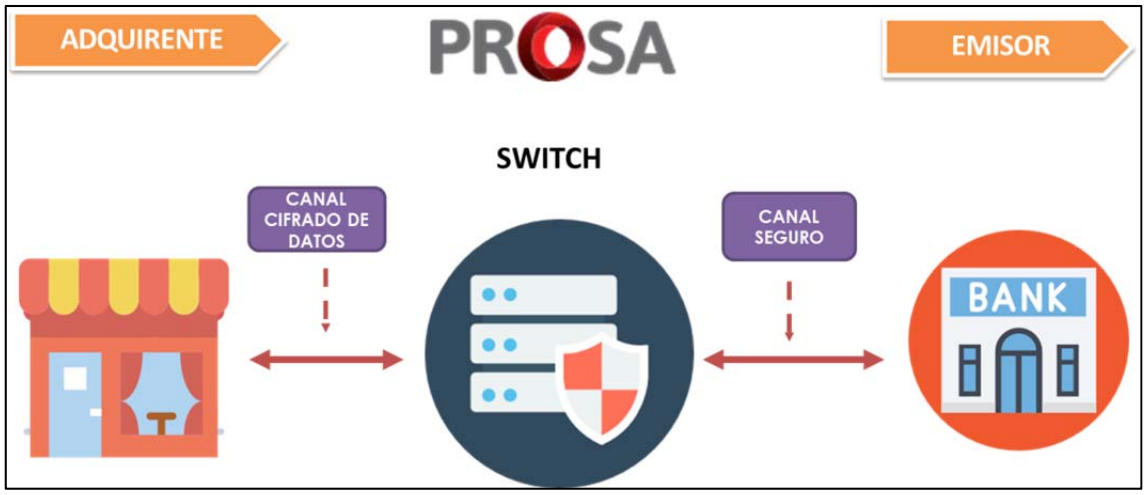

Imagen 4. Esquema general de cifrado Emisor Adquirente. (Especificación ISO de Interredes, mensajes encriptados PROSA, 2016).

Hardware de Seguridad HSM's Thales

El HSM es un módulo de seguridad de hardware (Hardware Security Module), su principal función es el manejo de llaves criptográficas, además realiza operaciones de cifrado, descifrado y firma digital por medio de algoritmos como AES, 3DES, RSA, entre otras.

El HSM hace imposible que las llaves generadas por este puedan ser exportadas para su uso en diferentes dispositivos y que la transmisión de datos

esté protegida. En caso de que el equipo fuese abierto, toda la información que almacena se borraría automáticamente.

Manejo de Llaves de Encripción

La tabla 1 describe los tipos de llaves y sus definiciones utilizadas durante el procesamiento de cifrado bajo el método DUKPT.

Tabla 1.

Llave	Descripción	Uso
Tk	Llave aleatoria generada por la terminal	Se envía en la solicitud de inicialización de llaves
RSApub	Llave pública inyectada a la terminal	Cifra la llave aleatoria RSApub(Tk) en la solicitud de inicialización de llaves
RSApriv	Llave privada en HSM	Permite descifrar la llave RSApriv(Tk) en una inicialización de llaves
K0	Llave inicial de cifrado	Llave inicial de cifrado enviada a la terminal
Kn	Llave incremental de cifrado	Llave generada por la terminal para cifrar una transacción
BDK	Llave de trabajo	▪ Genera la llave K0 para una terminal. ▪ Recrea la llave Kn utilizada en una transacción
KSN	Cadena identificadora de la llave BDK	▪ Generada por la BDK, asociada a la llave K0 en una inicialización de llaves. ▪ Identificador que permite recrear la llave Kn utilizada en una transacción cifrada.

Definición de componentes utilizados DUKPT. (Lineamientos para cifrado de datos sensitivos PROSA, 2010)

Las llaves descritas son utilizadas bajo el estándar de Implementación de DUKPT, estas definiciones no sufren cambios en la implementación del método para el cifrado de datos sensitivos en PROSA, salvo la variante definida en el HSM para uso de datos sencitivos.

Descripción de Datos Sensitivos

Hoy en día todos los datos sensitivos del mensaje ISO viajan en claro desde el dispositivo punto de venta hasta el Host Adquirente. No existe cifrado de esta información. Dado que tampoco hay captura de NIP del tarjetahabiente, no ha sido necesario implementar ninguno de los esquemas de seguridad diseñados hasta el momento como los utilizados en cajeros automáticos. A continuación, se detallan los datos sensitivos importantes que viajan en claro.

Track 2

El Track 2 es un elemento del mensaje financiero que contiene información codificada en la banda magnética ubicada en la parte posterior de la tarjeta que origina la transacción y/o el CHIP de esta, excluyendo caracteres de control de redundancia longitudinal (LRC). El contenido de los datos del Track se especifica en la norma ISO 7813, Tarjetas de transacciones financieras (BASE24 Product Documentation, 2004), ver tabla 2.

El Track2 actualmente viaja en el campo 35 del protocolo utilizado en transacciones financieras ISO8385 con toda la información en claro, sin cifrar.

Tabla 2.

Campo	Formato	Descripción	Comentarios/ Posibles Valores
35. Track 2 Data	ANS V 2:37 Alfa Numérico String Longitud variable en las 2 primeras posiciones, 37 posiciones de datos como máximo	Contiene la información del track 2 almacenada en la banda magnética.	*No incluye los caracteres* *(sentinels) de inicio y fin del track2.* EL *Field Separator* es el carácter "=". Para el caso de tarjetas digitadas, el track2 se compone del número de cuenta y separador ("=") y la

			fecha de expiración en formato AAMM

Formato del campo 35 ISO8583. (BASE24 Product Documentation, 2004).

El siguiente es un ejemplo del campo 35, ver imagen 5. La longitud del Track2 en este caso es de 33 posiciones, aunque puede variar hasta 37, lo cual normalmente depende de la longitud del Trailer. El Trailer es la porción del campo 35 que contiene la información sensible. El campo 35 completo, incluyendo la longitud en este ejemplo es de 35 posiciones.

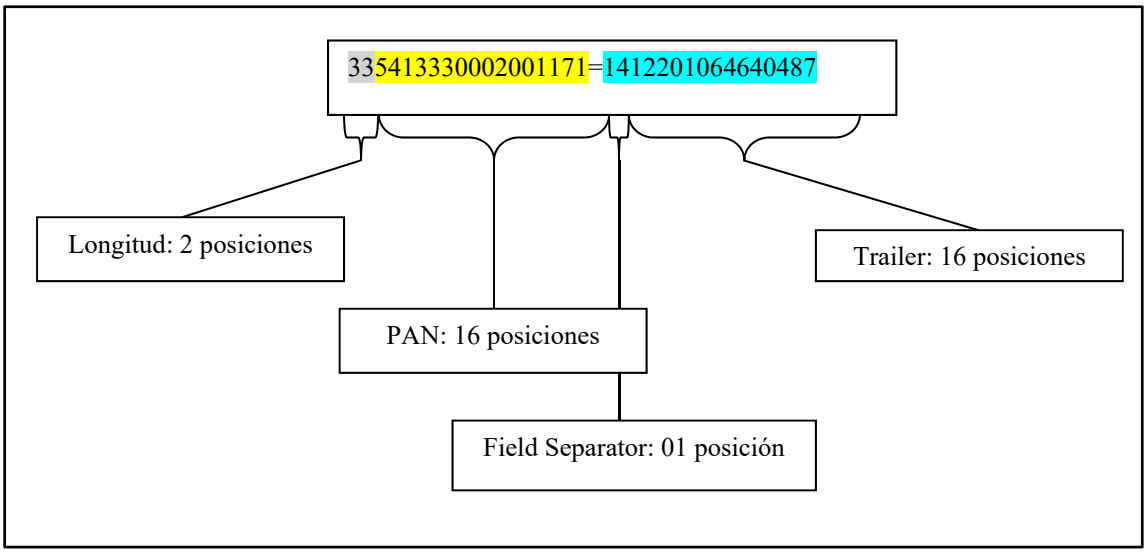

Imagen 5. Composición del Track 2 de la tarjeta. (BASE24 Product Documentation, 2004).

Código de Seguridad (CVV2 / CVC2)

El código de seguridad actualmente está impreso de manera legible en la tarjeta. Su ubicación y longitud depende del producto y marca (VISA, MASTRECARD, AMEX).

Actualmente no todas las tarjetas contienen este dato, por lo cual sólo es ingresado cuando la tarjeta sí lo tiene disponible. En la actualidad el ingreso de este valor se hace a través del teclado de la Caja Registradora Electronica

(ECR, por sus siglas en inglés, *Electronic Cash Register*)[9] y el envío al HOST Adquirente es realizado por medio del Token C0 del campo 63 del mensaje ISO.

Track1

El elemento Track 1 Data contiene la información codificada en la Banda Magnética de la tarjeta que se utiliza para la transacción, incluyendo el inicio y el final Sentinel y caracteres de control de redundancia longitudinal (LRC). El contenido de este elemento de datos se especifica en la norma ISO 7813, Tarjetas de transacciones financieras. El formato general de la información en estos datos se muestra a continuación (BASE24 Product Documentation, 2004):

- Inicio de sentinel (%)
- Código de formato (B para tarjetas de crédito es el único código de formato definido)
- Número de cuenta principal (PAN), justificado a la izquierda (hasta 19 dígitos)
- Separador de campos (^)
- Código de país (si está presente, 3 dígitos)
- Nombre (hasta 26 caracteres)
- Separador de campos (^)
- Fecha de vencimiento (YYMM)
- Código de servicio (si está presente, 3 dígitos)
- Datos discrecionales (hasta 21 caracteres)
- Fin de sentinel (?)

[9] Una caja registradora electrónica (ECR) es un sistema diseñado para permitir que los productos se vendan en un punto de venta. Las cajas registradoras electrónicas ayudan a los grandes minoristas a realizar un seguimiento de las ventas, minimizar los errores de registro, recopilar datos de inventario y mucho más. (Technopedia, 2017).

- Caracteres de comprobación de redundancia longitudinal

El Track1 es un dato que en la actualidad es requerido sólo por algunos emisores, en particular por American Express. Este dato viene grabado sólo en la banda magnética. En el CHIP EMV sólo hay porciones separadas del Track1, tales como el Track1 Discretionary Data, el Cardholder Name (Tag 5F20), pero NO el dato equivalente en su totalidad, como sucede con el Track2. El dato hoy en día viaja ya sea en el campo 45 (estándar) o como parte de un Token de usuario del Campo 63, o simplemente no es enviado al Host.

Solución desarrollada

Los mensajes ISO deberán tener cambios para soportar nuevos campos Cifrados e información de control, así como para eliminar el actual flujo de información sensitiva en claro. A continuación, se describen los requerimientos funcionales para los dispositivos puntos de venta, así como la implicación de llevar estos al simulador de mensajes ASSET.

Almacenar e informar los datos actuales de la Terminal Punto de Venta

El Host Bancario requerirá conocer el estado actual de cada Terminal Punto de Venta, el cual se hará llegar usando el Token ES de la mensajería ISO. Esta información le servirá al Host para determinar si la información sensible viajará cifrada, si debe requerir actualización de llaves, si debe actualizar BINES, o incluso para identificar la versión de aplicación y el Punto de Venta que está operando, en caso de haber problemas.

Los datos que el script debe entregar simulando la funcionalidad original de la Terminal Punto de venta para que el Host los pueda recibir en el Token ES del campo 63 del mensaje ISO son:

Versión Software. Este corresponde a la versión que el proveedor ha instalado en el Punto de Venta, esta no debe exceder 20 posiciones, para el caso del simulador se establecerá una genérica que identifique las transacciones son generadas por el simulador ASSET.

Serie de la Terminal. Corresponde a la parte numérica (o caracteres hexadecimales) que identifiquen a un dispositivo en particular, sin caracteres especiales. Este dato es tomado desde al sistema operativo o del *firmware* del dispositivo. Para el simulador se define uno arbitrariamente, pero se considera mantener el campo abierto para el caso en que se requiera simular la operación de un dispositivo real. No debe exceder a 20 posiciones.

Configuración de Cifrado. Este es un indicador de una posición, que debe indicar si la Terminal Punto de Venta está o no preparada para hacer cifrados; es decir, si ya hizo su primera inicialización de llaves de manera satisfactoria. Cada vez que se le vuelva a cargar la aplicación, la terminal debería quedar configurada para NO cifrar. Los valores del campo son:

0 – Configurado para NO cifrar. (aún no ha hecho su primera inicialización de llaves).
5 – Activo para Cifrado de Datos Sensitivos con DUKPT (primera inicialización de llaves hecha).

En el caso de Asset, este dato se registra en una variable con valor inicial igual a 0 y este dependerá del estado en que el script sea utilizado, ya sea para simular transacciones cifradas o sin cifrar, la duración de este dato estará presente solo durante la ejecución del script.

ID Tabla de BINES Locales Cargada. Es un identificador que la terminal almacena a partir de la primera tabla de BINES locales recibida (la cual puede

tener 0 bines informados). Si el Punto de Venta nunca ha cargado una tabla de BINES, entonces este indicador debe viajar en CEROS (“00000000”). Dentro del simulador este dato es procesado mediante una matriz definida que a la ejecución inicial del script se encontrará vacía.

ID Tabla de BINES Locales Informada. Es un identificador que la Terminal recibe en cada transacción y que devuelve para ser enviada en el Token ES. Sirve para que el Host Bancario pueda determinar si el Punto de Venta ha sido cambiado de comercio y actualizarle la tabla de BINES locales. Dado que en la simulación las tablas de bines siempre inician de forma vacía, este proceso no será tomado en cuenta por el simulador para realizar alguna acción.

Versión Tabla de BINES Locales. Corresponde a un número de dos dígitos hexadecimales que la Terminal recibe al momento de cargar la tabla de Bines locales: Este valor es almacenado y entregado al Host, con este se valida si hay una nueva versión de tabla para el comercio, en caso afirmativo, se transmite en la respuesta para ser cargado. El script, así como la Terminal, asumirá en su variable inicial que nunca ha cargado una tabla de BINES, por lo que define para este que el valor inicial debe ser “00”.

Nombre / Versión Llave RSA Pública para Firmas. Este dato viene precargado en la aplicación de la Terminal y corresponde a una cadena de 8 posiciones que informa cuál es la llave RSA de validación de firmas que la terminal tiene actualmente para con esto elegir la firma correcta. Este dato no es enviado al Host bancario, por lo que el simulador no tomara alguna acción en el que este dato sea diferente.

Bandera de Petición de Nueva Llave. La Terminal Punto de Venta utiliza este indicador para determinar si se encuentra activa para realizar cifrados o tomar las acciones necesarias para solicitar una llave nueva, cuando este es el caso

la Terminal internamente genera una llave aleatoria cada que este valor pasa a 0 y con esto queda en modo de carga de nueva llave. Por lo tanto, debe entregar una bandera que denote este estado. La bandera tiene los siguientes valores:

0 – NO está pidiendo nueva llave.
1 – Está esperando nueva llave desde el Host.

En el simulador este dato radicará en una variable, que inicialmente se encontrará en 1 (esperando nueva llave), con lo que el script determinará si ingresa o no a las rutinas para realizar cifrado de la información sensitiva. Este dato deberá ser censado previo a cada transacción que realice el script, tal como lo hace la terminal y a partir de ahí recurrir a las funciones indicadas para procesar la transacción a ser simulada.

Solicitar ó leer Datos Sensitivos dependiendo del modo de ingreso y cifrarlos usando 3DES con llave Simétrica DUKPT "Actual"

Una vez realizado de forma exitosa el proceso de carga de llave inicial, el Script simulando el comportamiento de la terminal deberá poder operar ahora en tres modalidades:

1.- Ingreso de datos. El script deberá continuar realizando las consultas a las bases de datos de tarjetas tal cual lo realiza en el modo sin cifrar pero deberá anteponer la condición del script para realizar la funciones de cifrado o no.

2.- Construir bloques de datos. Una vez con la información recibida de los datos de la tarjeta y del caso que se va a ejecutar, se validará el modo de entrada indicado en el campo 22 (Modo de entrada) con lo que el script

determinará la porción del Track 2 que deberá utilizar, así como si el Track 1 se encuentra presente o no. Una vez definido se procederá a la formación de bloques a ser cifrados.

3.- Cifrar usando 3DES usando la llave actual. Ya que se cuanta con la llave actual derivada mediante DUKPT y se identificaron los datos que está leyendo de la transacción, el script procederá a aplica Triple DES a cada porción del bloque de datos formado.

Esta información deberá ser enviada en los nuevos Tokens EZ y EY del mensaje ISO. La definición de estos es parte del Anexo 2.

Datos a Leer ó Solicitar de acuerdo al Modo de Ingreso. Dado que para el simulador no hay presencia física del CHIP o Banda Magnética como en la Terminal, la información a ser procesada será la utilizada en la base de datos de tarjetas definida previamente por el usuario del Script.

Los siguientes son los datos y estructuras de estos que serán leídos para cada uno de los modos de ingreso:

1. **Lectura de CHIP**. Sólo se dispondrá de la información del Track2 contenida en el CHIP. Por lo tanto, los datos a enviar al host bancario son:

    a. Bandera de CVV2 : **“A” → cvv2 no fue solicitado.**
    b. Longitud de CVV2 en claro : “00” → cero, porque no fue ingresado.
    c. Bandera de Track1 : “0” → no hay track1.
    d. Longitud de Track1 : “0000” → Cero, porque no hay Track1.
    e. Longitud de Track2 : “nn” → longitud del Track2 almacenado en el CHIP.

f. Modo de Lectura de la Tarjeta: “05” → Insertada.

2. **Lectura de banda o Fallback**. En este caso debería existir el Track2 o el Track1 (cuando NO hay presencia del track2), leídos desde la banda magnética, así como el código de seguridad (CVV2/CVC2/4DBC) tecleado por el operador de la caja en caso de haber sido solicitado por el PIN PAD. Los valores que DEBERIAN ser enviados son:
    a. Bandera de CVV2: “0” → no fue ingresado
        “1” → Si hubo CVV2.
        “A” → CVV2 NO solicitado por el PIN PAD.
    b. Longitud de CVV2 en claro: “00” (bandera “0” ó “A”) – “01” a “05” (otro caso)
    c. Bandera de Track1 : “1” → Hubo Track1 leído
        “0” → No hubo Track1 leído.
    d. Longitud de Track1 : “nnnn” → longitud del track1 multiplicado x 2. (cuando esté presente).
    e. Longitud de Track2 : “nn” → longitud del Track2 almacenado en el CHIP.
    f. Modo de Lectura de la Tarjeta: “90” → Deslizada – “80” → Fallaback

3. **Ingreso Digitado.** Se deberá pedir el PAN, opcionalmente la fecha de expiración (formato MMAA) y el código de seguridad (CVV2/CVC2/4DBC). Con el PAN y la fecha de expiración, el PIN PAD deberá conformar un Track2 “Parcial” antes del cifrado.

    a. Bandera de CVV2 : “0” → no fue ingresado
        “1” → Si hubo CVV2.
        “A” → CVV2 NO solicitado por el PIN PAD.

b. Longitud de CVV2 en claro: “00” (si no fue ingresado) – “01” a “05” (otro caso)
c. Bandera de Track1 : “0” → No hubo Track1 leído.
d. Longitud de Track1 : “0000” → Cero, porque no hubo Track1.
e. Longitud de Track2 : “nn” → Habitualmente “21”, resultado de concatenar el PAN, el carácter “D” y la fecha de expiración en formato (MMAA). Si la fecha de expiración NO fue solicitada por el PIN PAD, concatenar “0000”.
f. Modo de Lectura de la Tarjeta: “01” → Digitada.

Bloques a Cifrar. El bloque de cifrado de datos sensitivos debe contener el Track1, el Track2 y el código de validación (cvv2). Este bloque debe ser de una longitud fija y su longitud debe ser múltiplo de 16, por lo cual se utilizará el siguiente procedimiento para obtenerlo:

a) Obtener Porción Track2. Se toma el Track2 de la variable Fld35_Track_2 del Script adquirente completo o “parcial” (en caso de digitadas), se substituye el signo ‘=’ por una ‘D’ y se rellena con ‘F’s a la derecha para completar 38 posiciones. Para un track2 en claro: 32541333000200l171=141220106464048, el bloque resultante sería:

5413330002001171D141220106464048FFFFFF

b) Obtener Porción CVV2. Está porción se obtiene tomando como base el valor del código de seguridad tecleado y rellenando con el carácter ‘F’ a la derecha hasta completar 10 posiciones. Si consideramos que el cvv2 = 654, entonces seria: 654FFFFFFF

c) Porción Track1. Se convierte el Track1 a su valor hexadecimal (razón por la cual la longitud queda multiplicada x 2). Por ejemplo, si tenemos el siguiente Track1:

076B5177126237014591^CLIENTE/EL
^14011010000000000000701
000000

Los datos en negrita son la longitud del Track1 real y no se toman en cuenta para el cifrado, pero son los que se deben retornar multiplicados x 2 en el campo "*Longitud de Track1*". En este caso, se retornará el valor 76 * 2 = "0156".

Una vez obtenido el valor hexadecimal se debe rellenar con 'F's a la derecha hasta completar 160 posiciones.

```
42353137373132363233373031343539315E434C49454E54452F454C202
020202020202020202020202020205E313430313130313030303030303030
303030303030303037303130303030303030FFFFFFFF
```

Bloque Final Antes de Cifrado. Una vez obtenidas las porciones, estas se concatenan, quedando siempre primero la porción del Track2 (azul), luego la porción del CVV2 (en rojo) y la porción de Track1 al final (en amarillo). Las primeras dos porciones siempre deben estar, la tercera es opcional y siempre debe estar cuando el modo de ingreso sea 90 (deslizada) o de 80 (fallback).

```
5413330002001171D141220106464048
FFFFFF654FFFFFFF4235313737313236
323337303134353931 5E434C49454E54
452F454C202020202020202020202020
202020205E313430313130313030303030
303030303030303030303037303130303030
30303030FFFFFFFF
```

Aplicación del Cifrado con la Llave 3DES "Actual". Para aplicar el cifrado, se construyen bloques de 16 posiciones hexadecimales dividiendo el Bloque Final Antes de Cifrado. Para el siguiente ejemplo, los bloques a obtener son 13 en total:

```
5413330002001171D141220106464048
FFFFFFFFFFFFFFF4235313737313236
32333730313435393 15E434C49454E54
452F454C20202020202020202020202020
202020205E313430313130313030303030
303030303030303030303030303730313030
30303030FFFFFFFF
```

Conjunto de bloques a cifrar por Asset mediante la función “encrypt using des into” utilizando la llave derivada actual.

```
B1:  5413330002001171
B2:  D141220106464048
B3:  FFFFFF592FFFFFFF
B4:  4235313737313236
B5:  3233373031343539
B6:  315E434C49454E54
B7:  452F454C20202020
B8:  2020202020202020
B9:  202020205E313430
B10: 3131303130303030
B11: 3030303030303030
B12: 3030303730313030
B13: 30303030FFFFFFFF
```

Asumiendo que la llave diversificada y con variante aplicada (según algoritmo DUKPT) es el valor: 10AB93B59352E946B0844FC22ED1DE34, los bloques cifrados obtenidos aplicando 3DES son los siguientes:

```
B1:  29038990EA43E82B
B2:  736880129EF12D4D
B3:  2FF5DF178A5D51C4
B4:  925B6465F477CFFF
B5:  92231E01D447C90C
B6:  E87BD0AE0D70CE85
B7:  B106899DD8EBD9B7
B8:  8A4891171A3F489C
B9:  247EB34CB0B34A1A
B10: F562772D55612296
B11: A1AB1F5B7D10D032
B12: 118789BBAFA14F97
B13: AB3247DAAE2299F9
```

Bloques Cifrados a Retornar

Para que el Script Adquirente pueda enviar la información cifrada por Asset, se utilizan los Tokens: EZ y EY descritos previamente. En el Token EZ deben viajar los datos sensitivos cifrados correspondientes al Track2 y Código de Seguridad (Cvv2). En el Token EY deben viajar los datos cifrados correspondientes al Track1. De acuerdo con esto, la distribución de bloques para cada Token sería la siguiente:

Token EZ_→ Bloques B1 al B3.

Para el ejemplo, los datos bloques presentes en el Token EZ deberían ser:

```
B1:  29038990EA43E82B
B2:  736880129EF12D4D
B3:  2FF5DF178A5D51C4
```

Lo cual resulta en el buffer:

```
"29038990EA43E82B736880129EF12D4D2FF5DF178A5D51C4"
```

Token EY_→ Bloques B4 al B13.

Para el ejemplo, los datos bloques cifrados presentes en el Token EY deberían ser:

```
B4:  925B6465F477CFFF
B5:  92231E01D447C90C
B6:  E87BD0AE0D70CE85
B7:  B106899DD8EBD9B7
B8:  8A4891171A3F489C
B9:  247EB34CB0B34A1A
B10: F562772D55612296
B11: A1AB1F5B7D10D032
B12: 118789BBAFA14F97
B13: AB3247DAAE2299F9
```

Lo cual resulta en el siguiente buffer de 160 posiciones:

```
"925B6465F477CFFF92231E01D447C90CE87BD0AE0D70CE85B106899DD8EBD9B
78A4891171A3F489C247EB34CB0B34A1AF562772D55612296A1AB1F5B7D10D03
2118789BBAFA14F97AB3247DAAE2299F9"
```

Implementar DUKPT como método de Administración de Llaves para el cifrado de Datos Sensitivos.

Con el fin de liberar al Host Bancario de la responsabilidad de llevar una base de datos actualizada con las llaves que cada dispositivo posee, se implementa DUKPT como método de administración de llaves donde el Punto de Venta y en su caso el Script Adquirente de ASSET deberán cargar una llave inicial, la cual es diversificada a partir de una llave maestra existente en el Host Bancario. Esta llave inicial es almacenada en memoria del script y Asset aplicando la derivación genera llaves “hijas”, que son las que van cifrando la información y que una vez usadas son borradas de memoria. El Host Bancario identifica la llave “hija” utilizada a través del KSN, que es una cadena que el dispositivo debe almacenar, actualizar e informar al Host en cada transacción.

A continuación, se describen los componentes del método DUKPT y su implementación en ASSET. Como referencia de definición y funcionamiento del método en Dispositivo Punto de Venta y en el Host Bancario, se incluye el Anexo 4, donde se menciona la parte de implementación del método en el host bancario y en el dispositivo derivado de la norma ANSI X9.24-1-2017[10].

Descripción del KSN

El KSN (Key Serial Number) es una cadena de 20 posiciones hexadecimales usada por el método DUKPT para que el Host Bancario pueda recrear la llave con la cual un Dispositivo o Script cifrará la información en una determinada transacción, esto a partir de una Llave Base de Derivación (BDK) almacenada en el HSM (módulo encriptor) del Host Bancario, ver Imagen 6.

La norma ANSI X9.24-1-2017 Parte 1 define el algoritmo DUKPT para 3DES y solo describe el uso de los 21 bits del Transaction Counter (Ecounter),

[10] Retail Financial Services Symmetric Key Management Part 1: Using Symmetric Techniques. (ANSI X9.24-1, 2017)

el resto de los bytes puede ser definido de acuerdo con la necesidad particular de la solución para la cual se está implementando. El siguiente diagrama presenta la estructura genérica del KSN.

El KSN está pensado para que el *Host Bancario* pueda combinarlo con la llave BDK y obtener así la llave inicial que le cargará a un dispositivo. Por lo tanto, por cada llave inicial que el Host Bancario vaya a generar, deberá construir un KSN único definiendo un valor de *Key Set Identifier* y un *Key Serial Number*. Esta combinación de valores deberá ser única, dado que el *Ecounter* inicial siempre estará en CEROS.

Una vez que el Host Bancario genera la llave inicial del dispositivo, ésta debe ser cargada en el Punto de Venta junto al KSN utilizado para la diversificación. Cada vez que el dispositivo realice un cifrado de datos, debe incrementar el valor del Ecounter y enviar el KSN actualizado al Host para que éste pueda recrear la llave “actual” utilizada en el cifrado y descifrar los datos para su procesamiento.

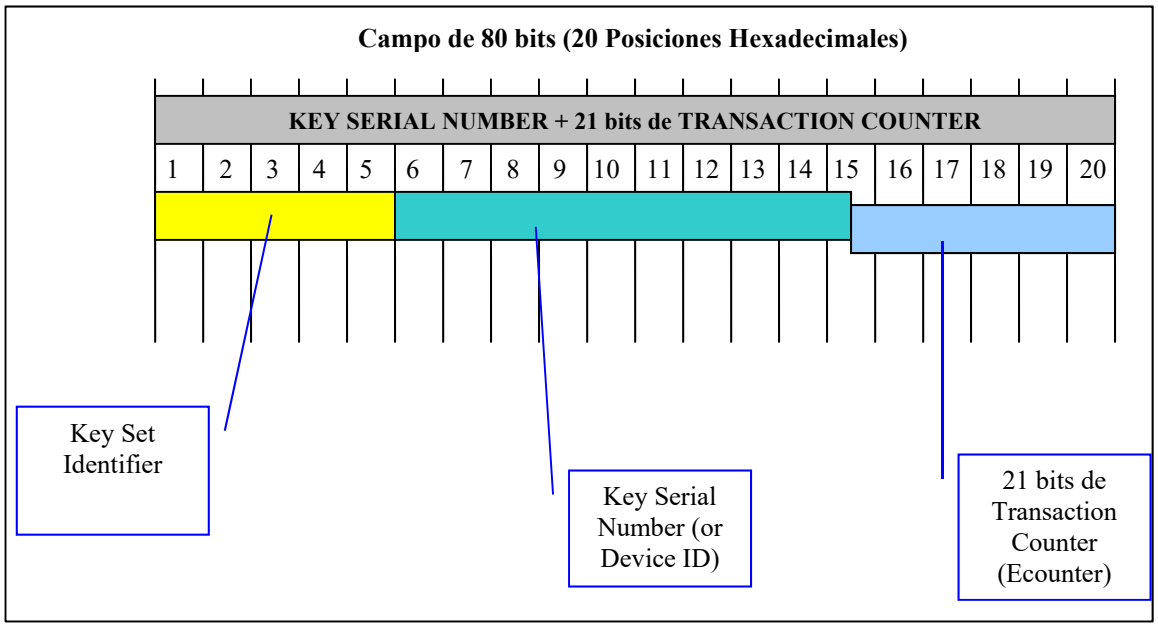

Imagen 6. Estructura del KSN. (Lineamientos para cifrado de datos sensitivos PROSA, 2010)

Algoritmo de DUKPT en el Script Adquirente de ASSET

Áreas de Almacenamiento

Serial_Number (59 bits). Almacena los 59 bits correspondientes a los datos del dispositivo. Asset manejará un dato constate que identificará que las transacciones estas siendo realizadas con un simulador y no con un punto de venta físico.

Encryption_counter (21 bits). Contador de cifrados, este contador puede saltar algunos valores, ya que no puede tener más de 10 bits encendidos.

Cipher_counter (7 dígitos decimales). Este contador sirve para almacenar el número de cifrados que ha realizado el script.

Key_serial_number (concatenación de Serial_Number y Encryption_counter). 80 bits (10 bytes).

Future_Key (21 registros). Estructura de datos de 21 registros de 34 dígitos decimales cada uno.

LRC "Longitudinal Redundancy Check" (2 dígitos hexadecimales). Determina si el registro es válido para usarse.

- Key_left_half → 16 dígitos hexadecimales
- Key_right_half → 16 dígitos hexadecimales

Current_pointer (apuntador de 4 dígitos hexadecimales). Contiene la dirección del registro en Future_key que va a ser usada en operaciones criptográficas.

Shift_register (21 bits). Registro de 21 bits numerados del 1 al 21 de izquierda a derecha. Este es usado para seleccionar un registro en Future_key indicado por la posición del bit encendido.

Shift_register_padd (64 bits, 8 bytes).

- Crypto_register_1 (16 dígitos hexadecimales). Registro para almacenar datos.
- Crypto_register_2 (16 dígitos hexadecimales). Registro para almacenar datos.

Key_register. Estructura de datos para almacenar llaves.

- Key_register_left (16 dígitos hexadecimales).
- Key_register_right (16 dígitos hexadecimales).

B1..B13. Variables definidas en ASSET de entrada de Bloques de datos Sensitivos en claro a ser Cifrados.
T1..T13. Variables definidas en ASSET de 16 dígitos hexadecimales para almacenar los Bloques de Datos Sensitivos ya Cifrados

Ejemplos de datos cifrados bajo el método de DUKPT en ASSET

Para hacer el cifrado de Track1, Track2 y cvv2 utilizando el método DUKPT, consideremos los siguientes elementos:

KSN inicial: 0102012345678AE00000
BDK inicial: 00112233445566778899AABBCCDDEEFF
TRACK1 : 076B5177126237014591^CLIENTE/EL ^14
0110100000000000000701000000
TRACK2 : 32541333002001171=141220106464048
CVV2 : 592

Basado en las reglas de construcción de bloques descritas previamente, el bloque de datos en claro queda de la siguiente forma:

```
5413330002001171D141220106464048
FFFFFF592FFFFFFF4235313737313236
32333730313435393 15E434C49454E54
452F454C20202020202020202020202020
202020205E313430313130313030303030
303030303030303030303030303730313030
30303030FFFFFFFF
```

Los bloques en claro quedan de la siguiente forma:

```
B1:  5413330002001171
B2:  D141220106464048
B3:  FFFFFF592FFFFFFF
B4:  4235313737313236
B5:  3233373031343539
B6:  315E434C49454E54
B7:  452F454C20202020
B8:  2020202020202020
B9:  202020205E313430
B10: 3131303130303030
B11: 3030303030303030
B12: 3030303730313030
B13: 30303030FFFFFFFF
```

Ahora aplicamos el algoritmo de DUKPT con el KSN inicial y el BDK inicial para calcular la variante de llave de cifrado, consideremos que el número de cifrado no corresponde con el contador en el KSN. Para la cantidad de cifrados tendremos un contador secuencial diferente al ecounter.

NOTA: Los ejemplos sólo son calculando sobre el Bloque B1.

1. **Para el cifrado número 1 y Ec = 2**

```
KSN Actual     =    0102012345678AE00002
VARIANTE PARA CIFRADO POS 20 = 10AB93B59352E946B0844FC22ED1DE34
Dato en claro =     5413330002001171
Dato cifrado  =     29038990EA43E82B
```

2. **Para el cifrado número 100 y Ec = 101**

```
KSN Actual     =    0102012345678AE00065
VARIANTE     PARA     CIFRADO     POS     21                =
9B12B5555BE0B56047A173B432672310
Dato en claro =     5413330002001171
Dato cifrado  =     DAE1DFB07A0CE9E9
```

3. **Para el cifrado número 1000 y Ec = 1001**

```
KSN Actual     =    0102012345678AE003E9
VARIANTE     PARA     CIFRADO     POS     21                =
ACD4D3C82DA384A836733BA17A227CE4
Dato en claro =     5413330002001171
Dato cifrado  =     3DF8BBB266C41AC0
```

4. **Para el cifrado número 10000 y Ec = 10096**

```
KSN Actual     =    0102012345678AE02770
VARIANTE     PARA     CIFRADO          POS     17           =
BB0205C0570C43117DFEB849F2AB2189
Dato en claro =     5413330002001171
Dato cifrado  =     8E68894B31D7BFF8
```

5. **Para el cifrado número 100000 y Ec = 115313**

```
KSN Actual     =    0102012345678AE1C271
VARIANTE      PARA      CIFRADO           POS      21           =
3EAEE738A50BFC95A25F6BF65B8AF42F
Dato en claro =     5413330002001171
Dato cifrado  =     72A86B75C0650C79
```

Mantener e Informar el Contador Real de Cifrados

Aunque el algoritmo DUKPT define un contador de cifrados como parte del KSN con el que se identifica la “llave actual”, éste no necesariamente es incrementado de uno en uno, por lo cual es necesario que el Punto de Venta lleve un contador verdaderamente secuencial, para que el Host Bancario pueda determinar si ya conviene hacer una actualización de llaves. Una vez actualizada la llave, este contador debe ser reseteado.

Tal como se ha comentado, el Ecounter manejado en el KSN no indica cuántos cifrados reales se han hecho luego de que el dispositivo fue inicializado con una nueva llave, ya que el Ecounter no se incrementa necesariamente de uno en uno.

A diferencia de la forma en que las Terminales Punto de Venta y los Pin Pads mantienen un Contador Real de Cifrados, en Asset este contador sería almacenado en una variable que estará disponible únicamente durante el tiempo de ejecución del script. Sin embargo, el procesamiento de la información de durante la ejecución de este deberá ser operada de la misma forma que lo hacen los dispositivos como se describe a continuación.

1. Deberá entregar actualizado el contador en cada transacción, para que el Host Bancario pueda determinar si corresponde o no instruir una actualización de llaves.

2. Este contador real de cifrados debe incrementarse siempre de uno en uno cada vez que haya terminado de cifrar TODOS los bloques de una transacción (desde el B1 al B13).

3. El contador debe ser incrementado, aunque la transacción NO haya sido aprobada.

4. El contador debe ser reseteado cada vez que se haya completado una "Inicialización de Llaves"

A continuación, se presenta una relación entre el valor del Ecounter del KSN versus el Contador Real de Cifrados.

Contador Real	Ecounter
1	2
100	101
1,000	1,001
10,000	10,096
100,000	115,313

Carga Remota de Llaves usando cifrado RSA con Llaves Públicas

Dada la cantidad de dispositivos instalados en el mercado, se requiere que el Punto de Venta y en nuestro caso el script que lo simula, pueda recibir la llave DUKPT "inicial" a través de un mensaje desde el Host Bancario. Para poder realizar esta carga se requiere que el Punto de Venta maneje llaves RSA Públicas para cifrar las llaves que serán transmitidas.

La carga remota de la llave DUKPT Inicial está concebida usando cifrado con llave RSA Pública por parte del Punto de Venta. Asumimos en este punto que el dispositivo o script ya cuenta con la llave RSA Pública, ya sea instalada

con la versión de software o bien entregada por la caja, para Asset esta llave será almacenada en el archivo de llaves de encriptación key_data.dat. La imagen 7 representa el diagrama del esquema de carga remota de llave inicial:

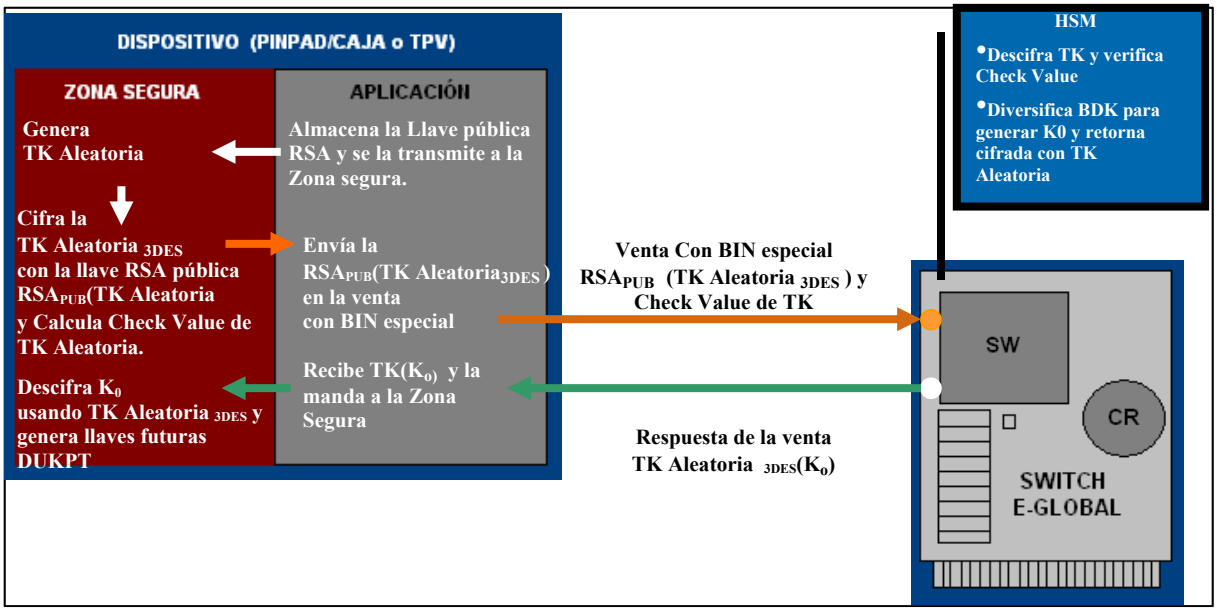

Imagen 7. Diagrama de la Carga Remota de la Llave DUKPT Inicial (K0). (Lineamientos para cifrado de datos sensitivos PROSA, 2010)

Descripción de Pasos:

1. El Host Bancario inicia una “Inicialización de Llaves” activando al Dispositivo con un comando especial y definiendo la Llave RSA Pública a utilizar, la cual debe ser de tamaño 2048 bits.

2. El Punto de Venta o Script genera una llave TK Aleatoria tipo 3DES de doble longitud, es decir, 32 posiciones hexadecimales.

3. EL Dispositivo cifra la llave TK Aleatoria usando la llave RSA Pública de Inicialización y con método de Padding PKCS V1.5[11].

4. El Dispositivo calcula el Check Value[12] de la Llave TK Aleatoria cifrando una cadena de 16 ceros.

5. El Punto de Venta o el Script que lo simula entrega al Host Bancario el cryptograma RSA_{Pub}(TK Aleatoria) y el Check Value.

6. El Dispositivo envía la solicitud al Host Bancario (Switch E-Global/PROSA en el ejemplo).

7. El Host Bancario descifra la llave TK Aleatoria usando la llave RSA Privada, verifica el Check Value, luego produce un valor de KSN único y con él diversifica la llave BDK Maestra para generar la nueva llave DUKPT Inicial (K0) a enviar al script o dispositivo.

8. El Host Bancario calcula el Check Value de la nueva llave inicial K0. Luego cifra la nueva K0 usando la llave TK Aleatoria y responde al Punto de Venta ambos datos: TK Aleatoria (K0) y Check Value de K0.

9. La Terminal Punto Venta o el Script Adquirente ejecutando las funciones de cifrado y descifrado de 3DES, descifra la llave K0 usando la llave TK

[11] En criptografía, PKCS # 1 es el primero de una familia de estándares llamada Public-Key Cryptography Standards (PKCS), publicada por RSA Laboratories. Proporciona las definiciones básicas y recomendaciones para implementar el algoritmo RSA para la criptografía de clave pública. Define las propiedades matemáticas de claves públicas y privadas, operaciones primitivas de cifrado y firmas, esquemas criptográficos seguros y representaciones de sintaxis ASN.1 relacionadas. (PKCS #1 V2.2, 2012)

[12] El KCV es el "Key Check Value" de la llave, es calculado suponiendo que los componentes clave son llaves 3DES y se obtiene cifrando una cadena de 16 ceros binarios. El KCV es los primeros seis dígitos hexadecimales del texto cifrado resultante. (Cryptomathic, 2017)

Aleatoria que él mismo generó, verifica el Check Value recibido y realiza el proceso de Inicialización de Llave de acuerdo al algoritmo DUKPT.

Ejemplo de Carga Remota de Llaves en Script Adquirente de ASSET

Pasos desarrollados.

1. El Script Adquirente genera una TK Aleatoria

   TK Aleatoria = 0123456789ABCDEF FEDCBA9876543210

2. El Script aplica un padding del tipo PKCS v1.5 sobre la TK Aleatoria, véase imagen 8.

Definición del PADDING PKCS #1:
El dato a ser cifrado con llave RSA Pública ó Privada debe ser construido de la siguiente forma:
00 BT PS 00 D

Dónde:
BT : Un byte indicando el tipo de cifrado.
01 – Cifrado con llave Privada 02 – Cifrado con Llave Pública

PS : Cadena de Padding, debe ser de al menos 8 bytes.
Para BT = 01 – La cadena debe estar formada por bytes “FF”x
Para BT = 02 – Bytes RANDOM diferentes a “00”x
D : La Data a ser cifrada.

NOTA: La longitud en bytes del bloque “*paddeado*” debe ser igual a la longitud del módulo de la llave a utilizar.

Imagen 8. Definición del PADDING PKCS #1. (PKCS #1 V2.2, 2012)

Para el ejemplo, la TK Aleatoria Paddeada con PKCS v1.5 para Llave RSA Pública =

```
00027F14F748E4804EF202B17BA24B3B3A1CCF7A0219913B98DF942EDCF239C3
67AF93C11B240EC6255924250291564D2C1BAA9CF0ACBB0D1E1988457C9C460C
A6EF187AC450BE419417E5AE6EB715469A6B47D156E9F8245435F1AAE885E306
A29FF1D8459A1DC42594C378A343C722E8D9254A4CC1ED2C515D64793052D1CA
```

269E0D77A305B2926ECCC89F021748DFE01DA7BDA88D9FB2B2A32E922357DAB9
8193FAAD426874D42CB47E204A9B09C683963621B9FC7D62A945EEF752AB83E6
63BC937F08F15AAFC6D766950997C29A2456BB10BC4A703F704DED252E276E44
EB52D8122496A4CBCAD3CD023DF82A000123456789ABCDEFFEDCBA9876543210

3. El Script selecciona (o recibe) la llave RSA Pública de 2048 Bits (Formato ASN.1 DER de la llave RSA en Hexadecimal ASCII).

RSAPUB =

30820122300D06092A864886F70D01010105000382010F003082010A02820101
00B4932D7A8D26B7DC5971758B31E0B72404248201E85222AD4F7ABF441A8B01
CE861BE54DD7BE35D4367B7D3EB87BAB8C08BDB0B624C94A79BE8715C4BFA08B
33B4C4686416F91AC4E5E50111D31FB4FD5D77CD51162DE153A506E720CBF7AE
4BECB5FB45803F927690A3B57965D068A107A55AB5E9D2F1C9FD1F021F2844FD
D4F655B9EC514E4338E4BCD9C818B92E051DC3B50611ADE7DBABA3195FA15C41
84DE3306E01FE2F51049F41C910B42E4531F4DBD4435D1F03930351EC68BA981
9655772DDD41353D800859778B47AFB03C51ABACB16EA0FAEADF7EAF86B16F87
BC231A61CEAC6A8D03B946911EEE010EADA4E06B1AC714C2D03C27DC4B4E9A07
DB0203010001

4. Posteriormente el Script cifra el bloque de la TK Aleatoria paddeada con PKCS V1.5 usando la llave pública RSA de 2048 bits. Entonces el cifrado RSAPUB(TK Aleatoria) =

1B249D03A980AA7A964E84A0D7883CA25FFB3957F3AB48C8A7988B934862D9BD
AB01118B7A20EE8FEA4B0E0AD3FB7DD09C14F154425375FC6F5DF1347F523AA0
8C627CE741CEF83770DB37C69AE1F65BA8084B8B9B3FC8E6545B56F9F35A9860
CD945B78AAC807BDD8BAAB26B16BEE704E2D1D99E3BDC5A056234EF49F78561A
8A7FB3C47759647CC3874EBBC8DDC1A85BE60FED769EBCDFADCDA9B60436B75B
F87E723DB62552916A6FB2B328368D3D7059973DBD1AF8E24E309FD6DC07684E
46800B5326D757A7F795807F0229BAC17D127F0FF31D3ABEABA68D0363806107
F7CE4A0B82D63515AEF79BFFEF88D8A28D6457CEC9445F074334683EE03739BB

5. También genera un CheckValue de TK Aleatoria.

Check Value = 3DES(“0000000000000000”, Tk Aleatoria) = 08D7B4FB629D0885

Check Value = 08D7B4 (Primeras 6 posiciones)

6. El PINPAD entrega la llave TK Aleatoria cifrada con la llave pública RSA y el checkValue de TK Aleatoria al comercio, quién los transmite al HOST BANCARIO por medio del Token “EW” del campo 63 del mensaje ISO.

Pasos del Host Bancario

7. El Host Bancario recibe TK Aleatoria cifrada con RSAPUB y el checkValue de TK Aleatoria. El HOST Bancario custodia una llave la llave privada RSA (Ejemplo de llave RSA Privada en Formato ASN.1 DER, desplegada en ASCII hexadecimal)

```
308204A40201000282010100B4932D7A8D26B7DC5971758B31E0B72404248201
E85222AD4F7ABF441A8B01CE861BE54DD7BE35D4367B7D3EB87BAB8C08BDB0B6
24C94A79BE8715C4BFA08B33B4C4686416F91AC4E5E50111D31FB4FD5D77CD51
162DE153A506E720CBF7AE4BECB5FB45803F927690A3B57965D068A107A55AB5
E9D2F1C9FD1F021F2844FDD4F655B9EC514E4338E4BCD9C818B92E051DC3B506
11ADE7DBABA3195FA15C4184DE3306E01FE2F51049F41C910B42E4531F4DBD44
35D1F03930351EC68BA9819655772DDD41353D800859778B47AFB03C51ABACB1
6EA0FAEADF7EAF86B16F87BC231A61CEAC6A8D03B946911EEE010EADA4E06B1A
C714C2D03C27DC4B4E9A07DB0203010001028201001 2C2CBD1D4C2760DCDB92D
06363B6BABB1467DBF66FCC99F8A076DC1CCA42E9EDF6E1C87D6E76B3E4E1ECC
676CD164845B102240FA5773866C962A5BD3B1016570DD8B1F03080490FA75AD
83C594A0C0462AC149306DB9E06FDFE4B992977C2365478F00AC8F6E4F253DED
07DCDD2751BBB2A1570F211C7FE8ED84D1E9E4291EFF6CCC27D4F474AA8CA439
0CECA3855F2129322CB8A5CB305EB2E669ED494BCF87A3477392F5EEF0DA7076
7F4EDA6E24E5FD8AD1DB7344E28510BE54C215C8698CA16A85C528CC7D3693C0
51F5FD5CFC0D1298A00468AE9F31EF288F6F8D5BFD78293835737C1B243B1630
A20B97A407216BE54CCB59B0C1984BF1850B6AAC4902818100E3B776D40ECE44
F6DDA66D6EA0059BE0EAC9112E0402FCDF9C2DDF509930AD1F80A31F57FE8B65
B53370AE8F178A6C8F8FBE4D98F4DA60A2BA578CF42FCD3E7B5A1CA680DC415C
DCE659383F78277423DC7F27565D472B8D5BF612D3E92BE30DCC9990F5440F62
40946E435D6BD1CBD7A41340E4DD78B630D7E965A13A65FBA702818100CB00C8
6420804EE76818E0EEE513E50E7C4813ECB373CE712E208DAE7E4AE502D7FC71
C854B7ECF189BCBAFA581D7DEAE1EB5E659369C4D55CD0D8C95B806E4ACDD318
F7DF027A7C24555573A6F3D6D497A9F4F584258C725C016E64CD1F3A8CDEE15A
0DDE0D646C82DC2631D73DD354BA0AD5BDBA24BFC35525B19E0CD848AD028181
00DA1950BD6004DD7697EA1BABB5A6499ACD2490C8C2DCDDB7898D41F29F1648
EA0039BBFE7A74A6903B597446E6A414C0174B8C64B0372C110F6A653D473F4B
D5B703F3D13DA378BEB5FCC3EBBE38037B89675B94D33824157AFB9F22C993D8
```

```
1ED2544ECB2A4995B006B9E8D9643807055D477782FC3FEE37AFC4022BBE9408
C70281810096B844102319607BEB0BDDA5412B9E3B3D06FED415007C5C07D55F
9C18645ED7D7A8B489FA6C7C5D70D85132A1CD73B9BA76D8252E67AF1E82C504
CFAB4215A13063F4FBC3F5C11EFE92E4D8F331C365201DADA3C3B3652D5B262C
E266713335781D74912B366243367D61564BEAB94704C21AA3BCDAA00DD4E021
99A90BC41902818050E5AE3E8DE3CF630196F0229CF917AF20FD65DA36F4E4A3
1ABD31242CD9A0E8513159BCE669C2CEFED0993726DEC6F5901DD26743E2F4C3
D71E6B0619687C4519FF3C79EE31A60ACD88D3DB379A790D52D1BFF65D935566
C0B3F5D71C36A92B66BB3BF2B70E2AE6D9941E52E78153124A72AB1CE50F7347
94D295BDF51E6945
```

8. Descifra RSA_{PUB} (TK Aleatoria) con la llave privada RSA de 2048 bits con algoritmo de padding PKCS 1.5.

   TK Aleatoria = 0123456789ABCDEF FEDCBA9876543210

9. Comprueba el checkValue de la llave TK Aleatoria, si no coincide, retorna un código de error y termina el proceso de generación de nueva llave, de lo contrario continua con el siguiente paso.

10. Crea un KSN0 para la nueva llave, que para nuestro ejemplo es:

    KSN0 = 01020123456789E00000

11. El Host Bancario diversifica la llave BDK Maestra usando el nuevo KSN0 y obtiene la nueva llave inicial K0 que se enviará al dispositivo:

    BDK Maestro = 00112233445566778899AABBCCDDEEFF
    KSN0 = 01020123456789E00000
    K0 = B1F70AC8B1281F80ADC6EAE4C3039FE9

12. El Host Bancario genera el Check Value del K0.

    Check Vaue K0 = 3DES(Cadena de Ceros, K0) = 45E00ADFAD962ABF

13. El Host Bancario cifra el K0 usando el TK Aleatorio.

    TK Aleatorio (K0) = 3DES(K0, Tk Aleatorio) = 9C8AC9BD70500C6F 354E87D3D833A920

14. Finalmente, los datos retornados por el Host Bancario son:

    TK Aleatorio (K0) = 9C8AC9BD70500C6F 354E87D3D833A920
    Check Value = 3DES("0000000000000000",K0) = 45E00ADFAD962ABF
    Check Vaue K0 = 45E00A (Primeras 6 posiciones)

Validación de Firma de Llaves RSA Públicas

Como la Terminal Punto de Venta podrá solicitar "Inicialización de Llaves" a cualquiera de los Host Bancarios con los que tiene contrato el comercio, entonces deberá poder recibir la llave RSA Pública que le permitirá llevar a cabo el proceso de Carga Remota de Llaves con el Host Bancario seleccionado. Sin embargo, para que el modelo sea seguro, el dispositivo deberá exigir una firma asociada a la llave RSA Pública que se le está entregando. La firma de esa llave deberá ser validada por el Dispositivo usando una llave RSA Pública para validación de firmas. Si la firma no concuerda, el proceso deberá ser abortado por la Terminal.

El modelo de Carga Remota de Llaves requiere que el Dispositivo cifre una llave TK Aleatoria con la llave RSA Pública del Host Bancario al cual se le solicitará la generación de la nueva llave DUKPT Inicial (K0) a cargar en este.

A primera vista, la llave RSA Pública podría formar parte de la aplicación cargada en el Punto de Venta; sin embargo, existe la necesidad de que el comercio pueda solicitar la transacción de Inicialización de Llaves a cualquier Host Bancario con el que tenga convenio de operación. Esto implica que la

llave RSA Pública a utilizar depende de a cuál Host Bancario se enviará la solicitud de Inicialización de Llaves.

Para que este modelo sea seguro y un defraudador NO pueda entregar a la Terminal una llave RSA Pública fraudulenta, se requiere que la caja almacene las llaves RSA Públicas de los diversos Host Bancarios pero firmadas con una llave de autenticación, para que el dispositivo pueda validar que la llave que recibe es NO es fraudulenta.

Dado que para el simulador en esta primera etapa de implementación será operando únicamente para proyectos de certificación utilizando al Switch de PROSA como Host Bancario, se asume que las llaves RSA que ASSET utilice se encuentran validadas por lo que el proceso de validación de firmas no será aplicado por el simulador.

Cargar de tabla de BINES para determinar si se deben cifrar los Datos dependiendo del prefijo de la tarjeta

La terminal deberá cargar una tabla de BINES LOCALES del comercio; es decir, aquellos prefijos que el comercio no enviará al Host Bancario y que por tanto requieren los datos sensitivos en claro. Los bines que se pretende integrar en esta tabla son: marcas propias emitidas por el propio comercio, tarjetas de lealtad y emisores que no pasan por los Switches Bancarios tradicionales, etc.

La tabla de BINES LOCALES que le corresponde a un dispositivo será controlada por el Host Bancario. Esta tabla será respondida en línea por el Host Bancario a la Terminal. La tabla de BINES LOCALES tiene un tamaño limitado, definido por el mensaje ISO, además para seguridad, viajará cifrada para evitar que el comercio la pueda alterar o ingresar una fuera de los mecanismos definidos.

Debido a que esta es una operativa más enfocada al funcionamiento y administración propia del software y hardware la terminal no fue incluida como parte de las funciones de simulación desarrolladas en ASSET.

Operar sin cifrado de datos mientras no haya inicialización de llaves

Para efecto de proporcionar una forma dinámica de implementar el uso de terminales con cifrado se definió que mientras el dispositivo no haya hecho su primera inicialización de llaves los datos los entregará en claro al Host Bancario, por lo tanto, se debía contemplar en la interfaz del simulador el flujo de los datos en sus 2 modalidades: en claro y cifrados. Para que el simulador pueda operar de manera correcta, se especificó que los comandos del script retornen lo siguiente:

- Indicador de si los datos a procesar están cifrados o no.
- Datos Sensitivos Cifrados ó bien Datos Sensitivos en Claro, más nunca las dos modalidades juntas.

Al simulador se le deberá especificar en qué campos deberá poner cada tipo de información, como se muestra en la tabla 3 a continuación

Tabla 3.

Dato	Estado	Campo ISO
Track1	En claro	No usado
Track2	En claro	Campo 35
CVV2	En claro	Token C0 en campo 63
Track1	Cifrado	Token EY en campo 63
Track2	Cifrado	Token EZ en campo 63
CVV2	Cifrado	Token EZ en campo 63

Campos sugeridos para el manejo de mensajería financiera con y sin cifrado. (BASE24 Product Documentation, 2004).

Queda a consideración del HOST Bancario el declinar la transacción o no, en caso de que el comercio envíe los datos sensitivos “en claro” y la configuración de cifrado en el Token ES viaja indicando que el Script está configurado para cifrar. Esta situación sólo podría darse en caso de un problema en la tabla de BINES Locales. Para simular este comportamiento será necesaria la edición manual de la información del Token ES dentro del archivo de casos a nivel adquirente de ASSET Acquirer_Test_Cases.

Calcular / Validar el CRC32 para campos cifrados

Como una medida para descartar problemas en el descifrado, se requiere que el Script envíe un valor de CRC calculado sobre los datos cifrados, de manera que el Host Bancario pueda detectar problemas de transmisión previo a intentar descifrar. Así mismo, para los datos que el Host envíe cifrados hacia el Script, también mandará el CRC32, para que Asset pueda detectar problemas antes de asumir que la falla está en el cifrado mismo.

En general los campos que viajen cifrados deberán tener un CRC32 asociado y calculado por la entidad que cifró la información, para que la entidad receptora (Script o Host Bancario) puedan validar que la información llegó íntegra a través de las diversas funciones del simulador y descartar así fallas que erróneamente podrían interpretarse como problemas en el descifrado. A

continuación, se presenta un resumen de los datos que tienen un CRC asociado, quién debe generar el CRC y quién lo debe validar, ver tabla 4.

Tabla 4.

Dato	Token de Campo 63	Generador del CRC32	Validador del CRC32
Track1 Cifrado	EY	ASSET	Host Bancario
Datos Sensitivos Cifrados (Bloque Track2 / CVV2 Cifrado)	EZ	ASSET	Host Bancario
Llave Aleatoria Cifrada <RSA Pub (TK Aleatoria)>	EW	ASSET	Host Bancario
Llave Nueva Cifrada <TK Aleatoria(K0)>	EX	Host Bancario	ASSET
Tabla de BINES Cifrada	ET	Host Bancario	ASSET

Campos indicados en la generación de CRC. (Lineamientos para cifrado de datos sensitivos PROSA, 2010)

NOTA: Todos los CRC serán calculados sobre el valor hexadecimal ASCII que se transmita.

A continuación, se detalla sobre qué porción se hará el cálculo del CRC32 en cada dato:

A) Track1 Cifrado:

Se debe aplicar sobre los Bloques B4 al B13 concatenados, detallados en el capítulo “Construcción de Bloques a Cifrar”, que resulta en 160 posiciones.

B) Datos Sensitivos Cifrados

Se debe aplicar sobre los Bloques B1 al B3 concatenados detallados en el capítulo “Construcción de Bloques a Cifrar”, que resulta en 48 posiciones.

C) Llave Aleatoria Cifrada

Se debe aplicar sobre el dato RSAPUB(TK Aleatoria), que debería resultar en 512 posiciones.

D) Llave Nueva Cifrada

Se debe aplicar sobre el dato TK Aleatoria(K0) detallado en el punto “Pasos del Host Bancario”, del subcapítulo “Ejemplo Carga Remota de Llaves”, que es de 32 posiciones.

E) Tabla de BINES Cifrada

Se debe aplicar la *Porción usada del BUFFER CIFRADO*, descrito en el subcapítulo “Descripción de Pasos para Descifrar tabla de BINES LOCALES”. La longitud de este dato es variable, es múltiplo de 16 y debe ser entregada por la CAJA quién a su vez la recibe en el Token ET.

Algoritmo de Referencia y Ejemplos

A modo de referencia se adjunta un trozo de código en ASSET que realiza el cálculo del CRC32 para la TK Aleatoria, Track 2 y CVV2, ver imagen 9.

```
//-----------------------------------------------------------
// Si fue encriptada correctamente la Llave TK Aleatoria,
// se calcula el CRC de la llave resultante
//-----------------------------------------------------------
  if (TK_Encr <> "")
      set CRC32_Input = TK_Encr
      generate crc for CRC32_Input into Tkn_EW_Msg.CRC32_Val
  endif


//-----------------------------------------------------------
// Se calcula CRC-32 para Track2 y CVV2 Encriptados y se
// establece valor en el subcampo Tkn_EZ_Msg.CRC32_EZ
//-----------------------------------------------------------
  if (Tkn_EZ_Msg.DATA_CIFRA <> "")
     generate crc for Tkn_EZ_Msg.DATA_CIFRA into Tkn_EZ_Msg.CRC32_EZ
  endif
```

Imagen 9. Código de generación de CRC de llave aleatoria TK, Track2 y CVV2 encriptados por el Simulador ASSET. (ACI Payment Testing, 2016)

Pruebas de simulación de datos sensibles cifrados en AASET

“Simulación es el proceso de diseñar un modelo de un sistema real y llevar a cabo experiencias con él, con la finalidad de aprender el comportamiento del sistema o de evaluar diversas estrategias para el funcionamiento del sistema (Shannon, 1988).”

A continuación se presentan los diferentes procesos desarrollados en el simulador ASSET de acuerdo a la integración del método de administración de llaves DUKPT.

Solicitud de Inicialización de Llaves

De acuerdo a los lineamientos de encripción de datos sensitivos, una vez generada la programación necesaria en los Scripts de ASSET HPDH e ISO8583, se integra una nueva transacción que corresponde al comportamiento de la terminal para realizar el proceso de Inicialización de llaves.

En el ejemplo la llave RSA pública cargada en el script es la siguiente:

RSA_Key_Pub =

```
D55A54CD8DD3232DAA97CA42B8316D44096F2BDADD9DEC4A3E6EA50CEEA5A21BD42DCA
1B3EA504A14FEF604C9EB6DB633C429FBE4A9137357DAB5463D822FBD81BBA48A1830F
5D64C0B92FB7166FCD2BC847CE7905EC49B42BB6AFA4B297A4E387BFF31B0084502A79
C1832156C2B60E1BA71502240CE63BD0B28EAF96FF6D23CE9F17D1E1A1E33DCC69005F
ED32676B0D94A3160C733B2473C2AECFFF587A7064C2A1D8568A91FE728A9D3138C739
65083C2737B8C153DEE99247A695123BC3FB73BD34DF67BCADCF4EB31D6BE3E2DEEE10
751BC4D5E1FD8B9795594C0559BEFC23FBABD3B50A01719849A129AF018A68C2F48F17
A0D2CF9A678A6C543CA9C3
```

Asset genera la Llave Aleatoria y la almacena en la variable TK_Aleatoria, la cual es cifrada bajo la llave RSA pública y enviada en el Token EW junto con el Check Value que será utilizado para comprobar su integridad en el Host de PROSA, de igual manera se envía el Token ES con datos de la terminal como se muestra en la imagen 11.

Para completar la solicitud de inicialización de llaves de acuerdo a los lineamientos de DUPTK dentro del Host, el Scritp envía en un mensaje de solicitud de transacción financiera (0200) los campos Fld03 (processing code) = 000000 y Fld04 (monto de la transacción) = 000000000000, ver imagen 10. Los campos adicionales corresponden a datos informativos de la transacción y la terminal.

Transacción de Inicialización de llaves

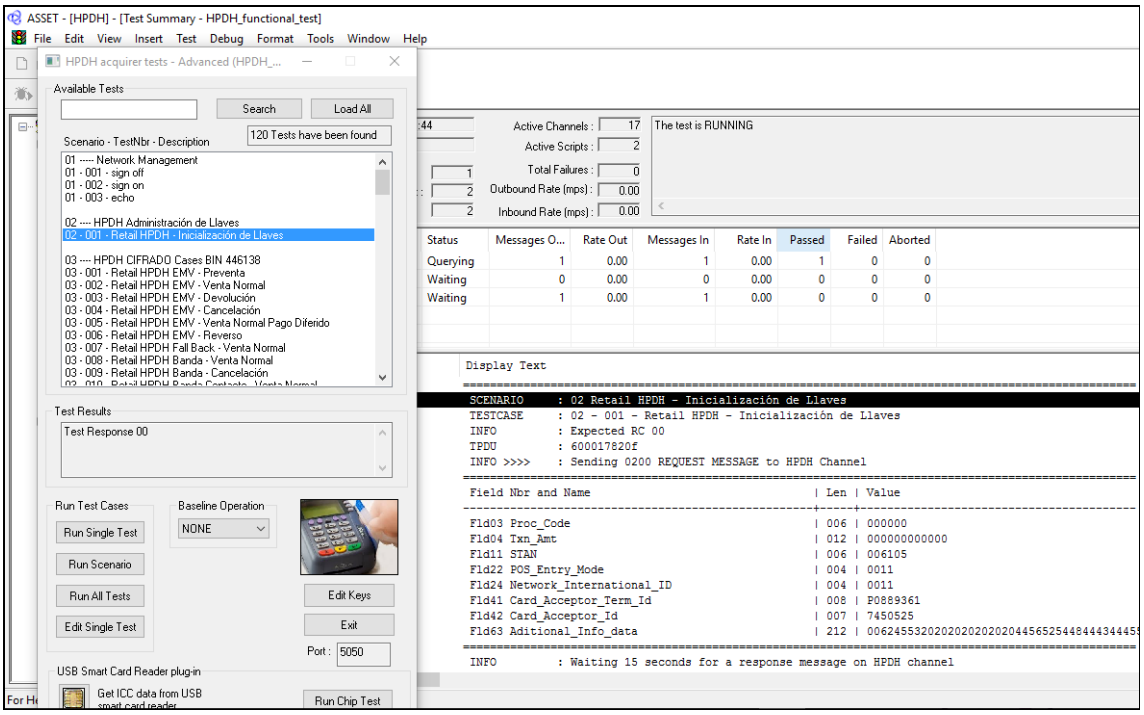

Imagen 10. Transacción de inicialización de llaves en el script HPDH de ASSET. (ACI Payment Testing, 2016)

Información de campo 63 con Tokens de solicitud ES y EW

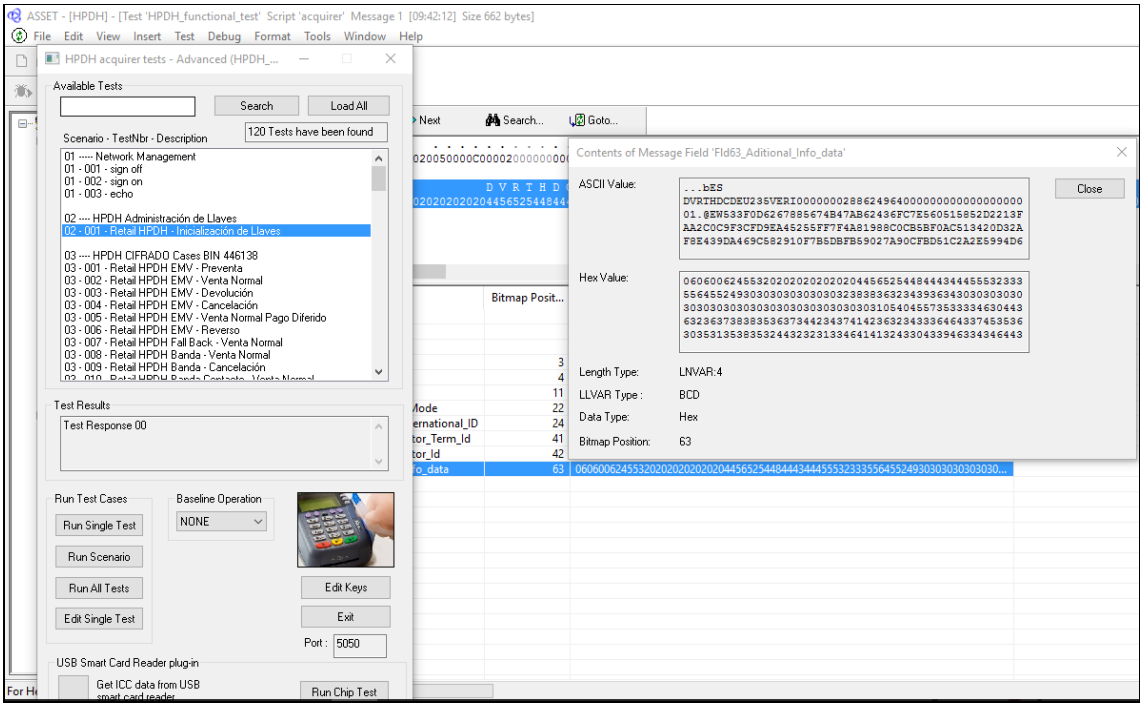

Imagen 11. Detalle de campo 63 con información de solicitud de llaves. (ACI Payment Testing, 2016)

Por su parte el Host de PROSA al recibir la transacción de generación de nueva llave, envía en primera instancia al módulo de seguridad HSM, el comando GI (Import a DES Key) la llave TK Aleatoria cifrada bajo la llave publica RSA y realiza el proceso de traslado de una llave DES cifrada bajo una llave pública hacia la llave LMK (llave maestra local) del módulo de seguridad de PROSA. Imagen 12.

Comando GI en el HSM de PROSA

```
 1130    281 17/08/08 09:41:57.37                          771884       2 -

      SOURCE P1H^RAM^10         ( PRO 67 )
      DEST   S1H^RAM10^01       ( STA 79 )

             0: 3033 3531 4749 3031 3031 3330 3030 3032 [0351GI0101300002]
            16: 3536 533F 0D62 6788 5674 B47A B624 36FC [56S?.bg.Vt.z.$6.]
            32: 7E56 0515 852D 2213 FAA2 C0C9 F3CF D9EA [~V...-".........]
            48: 4525 5FF7 F4A8 1988 C0CB 5BF0 AC51 3420 [E%_.......[..Q4 ]
            64: D32A F8E4 39DA 469C 5829 10F7 B5DB FB59 [.*..9.F.X).....Y]
            80: 027A 90CF BD51 C2A2 E599 4D6B 264B F7CA [.z...Q....Mk&K..]
            96: A0DA 4671 6309 EBA9 3F24 C29E 6CAA A85F [..Fqc...?$..l.._]
           112: 1B09 312A 1647 283D B230 5D71 0BA8 F2CA [..1*.G(=.0]q....]
           128: BCB4 B1D6 1FD9 2851 B3B8 3B0E 9DBB CD8B [......(Q..;.....]
           144: CEC1 AD9D E187 19FE 2CE8 304A E933 14E8 [........,.0J.3..]
           160: 5BDF 8326 50C5 5788 857E F79B E80D D825 [[..&P.W..~.....%]
           176: 3BFC B884 1DFE B8AD C317 20B0 2657 490E [;......... .&WI.]
           192: 74B8 0EC7 3361 B58B C09A 7626 5BA6 3CDB [t...3a....v&[.<.]
           208: C0D9 66D9 BA5B CE7C 8362 276E E055 E07F [..f..[.|.b'n.U..]
           224: 3809 9F0D 3ED6 28FA 89A3 809F C505 6909 [8...>.(.......i.]
           240: B26A E6A9 117D 3685 5D91 98D6 F56E 8BB0 [.j...}6.]....n..]
           256: 550E 7701 FD42 7643 20B7 3E92 FBBE 9E0B [U.w..BvC .>.....]
           272: AAD5 3B30 303B 3B55 31                  [..;00;;U1]
```

Imagen 12. Importación de llave generada por el Script Adquirente de Asset en el módulo de seguridad HSM del Host de PROSA. (Thales HSM 8000, 2002)

Si todos los datos fueron entregados y procesados de forma correcta en la solicitud al HSM, este devuelve como respuesta la llave aleatoria generada por el Script cifrada ahora bajo la llave LMK del HSM como se muestra en la imagen 13.

Comando de respuesta GJ desde el HSM

```
 1131     47 17/08/08 09:41:57.39                          771885       0 -

      SOURCE S1H^RAM10^01       ( STA 79 )
      DEST   P1H^RAM^10         ( %177777 )

             0: 3033 3531 474A 3030 5536 3930 4138 4336 [0351GJ00U690A8C6]
            16: 4546 3939 3937 3832 3830 4330 4433 4530 [EF99978280C0D3E0]
            32: 4437 4542 3832 4544 4432 4331 3646 45   [D7EB82EDD2C16FE]
```

Imagen 13. Respuesta del módulo de seguridad con llave TK Aleatoria cifrada bajo la llave LMK del HSM. (Thales HSM 8000, 2002)

En el caso de producirse un error durante la importación de la llave TK Aleatoria generando como resultado un valor diferente a “00” en el comando GJ, se entregará al script un mensaje de respuesta tipo 0210 (respuesta de transacción financiera) con el código de error en el campo Fld39 (código de respuesta) de acuerdo a lo definido en la tabla del anexo 3.

Una vez que la importación de TK Aleatoria generada por el script fue satisfactoria y el programa del Host de PROSA la recibe cifrada bajo la LMK, se envía el comando XC encargado de generar la Llave Inicial de Encripción IEK (por sus siglas en inglés, *Initial Encription Key*), tomando como dato la Llave Base de Derivación BDK (por sus siglas en inglés, *Base Derivation Key*) almacenada en el Host de PROSA, el Check Value de la llave TK Aleatoria y KSN asignado por el Host a la terminal para mantener el contador de derivaciones de la llave inicial, ver imagen 14.

Comando XC de generación de llave incial IEK en el HSM

```
1133     93 17/08/08 09:41:57.39                          771886       6 -

      SOURCE P1H^RAM^10          ( PRO 67 )
      DEST   S1H^RAM10^01        ( %177777 )

             0: 3033 3532 5843 5536 3439 4142 3933 4442 [0352XCU649AB93DB]
            16: 3343 4330 3539 4144 3030 3830 3341 4231 [3CC059AD00803AB1]
            32: 3930 3843 3330 3255 3639 3041 3843 3645 [908C302U690A8C6E]
            48: 4639 3939 3738 3238 3043 3044 3345 3044 [F99978280C0D3E0D]
            64: 3745 4238 3245 4444 3243 3136 4645 3030 [7EB82EDD2C16FE00]
            80: 3030 3030 3538 3030 3639 3536 36        [0000580069566]
```

Imagen 14. Comando XC de generación de llave inical IEK a partir de una llave maestra DBK. (Thales HSM 8000, 2002)

En respuesta a la solicitud enviada al HSM con el comando XC, este produce una respuesta con el comando XD (imagen 15), el cual contiene la llave inicial de encripción (IEK) cifrada bajo la llave TK Aleatoria generada por el escript así como el Check Value generado para su comprobación, véase la imagen 15.

Comando XD de respuesta de generación de llave en el HSM

```
1135     46 17/08/08 09:41:57.39                          771887       0 -

      SOURCE S1H^RAM10^01        ( STA 79 )
      DEST   P1H^RAM^10          ( %177777 )

             0: 3033 3532 5844 3030 4133 3835 4242 3239 [0352XD00A385BB29]
            16: 3336 3236 4241 4138 3345 3130 3837 4536 [3626BAA83E1087E6]
            32: 3344 4535 3435 3246 3544 3235 3237      [3DE5452F5D2527]
```

Imagen 15. Comando XD de respuesta con llave IEK cifrada bajo TK. (Thales HSM 8000, 2002)

Ya con la llave IEK generada y cifrada bajo la TK Aleatoria del script, el Host de PROSA procede a generar los Tokens ES y EW con información que el script utilizará para transaccionar utilizando cifrado en los datos sensitivos de las tarjetas, ver imagen 16.

Tokens de respuesta al Script HPDH en Asset

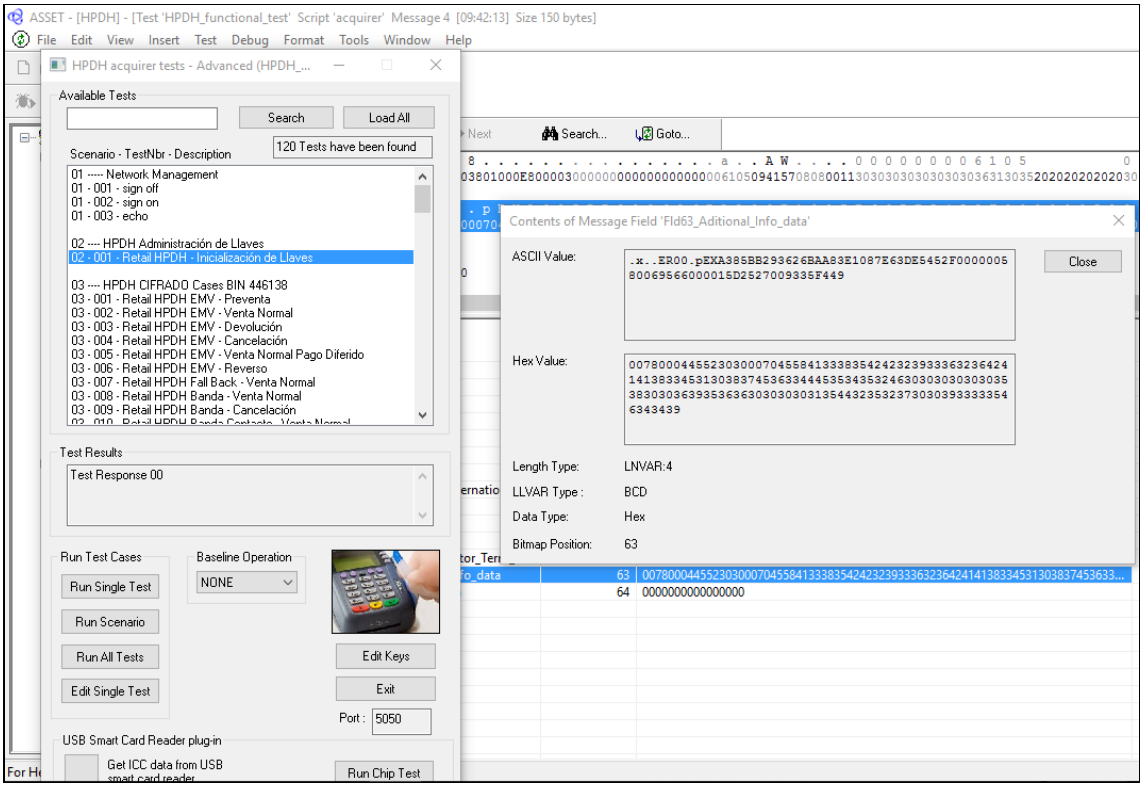

Imagen 16. Detalle de campo 63 con Tokens de respuesta ER y EX con IEK cifrada TK Aleatoria. (ACI Payment Testing, 2016)

Dado que a la llave IKE “A385BB293626BAA8 3E1087E63DE5452F” generada por el HSM es necesario desencriptarla mediante Triple DES utilizando la llave TK Aleatoria y posterirormente aplicar de nuevo 3DES con la variante de cifrado para uso de Encripción de Datos “0000000000FF0000 0000000000FF0000”, obtenemos la llave IKE “B24454972D5ACFFC 02C515E219FD8864” a utilizar por Asset para iniciar la encripción de bloques de datos con información sensible de las tarjetas, ver imagen 17.

Respuesta a transacción de Inicialización de llaves

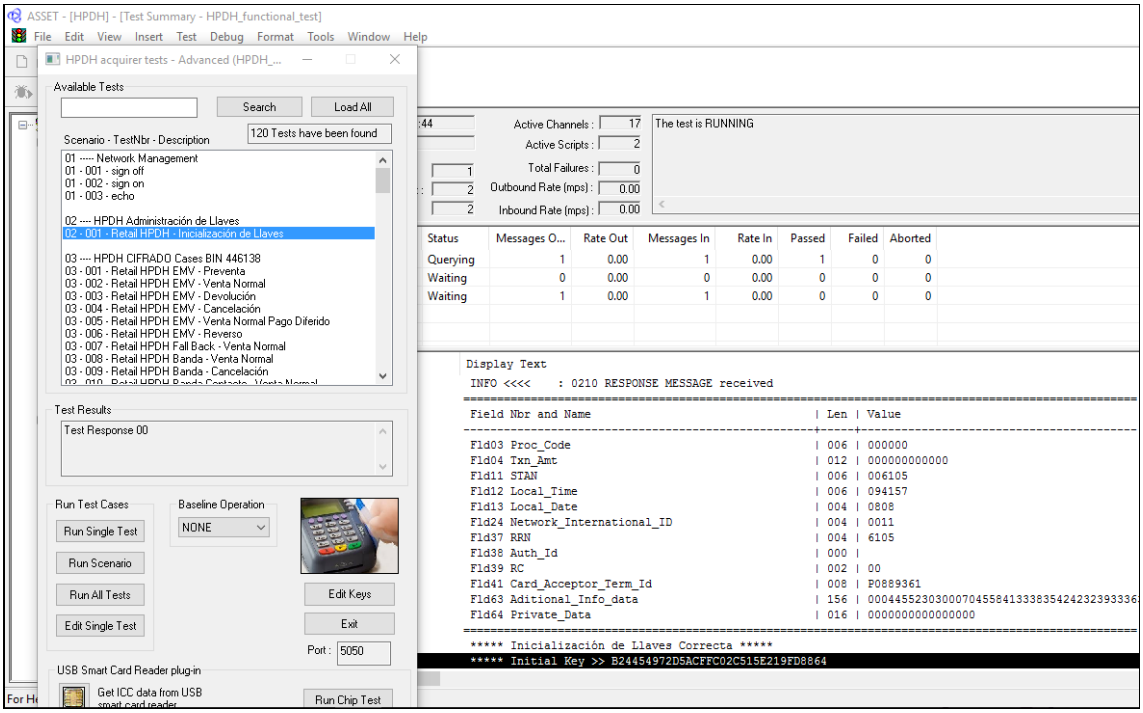

Imagen 17. Respuesta de Inicialización de llaves en hacía Script de ASSET. (ACI Payment Testing, 2016)

Transaccionando con Cifrado posterior a la carga de llaves

Realizado de forma exitosa el proceso de carga Inicial de Llaves, el script Adquirente enciende de manera interna un indicador con que internamente le indica que se deben ejecutar las funciones para aplicar Metodo de Administración de Llaves DUKPT, así como su respectiva derivación de llaves, incrementar los contadores de derivación y cifrar los datos sensitivos de las transacciones subsecuentes a realizar. Ver la imagen 18 con el ejemplo de venta normal con cifrado de datos.

Venta Normal en Script HPDH de ASSET

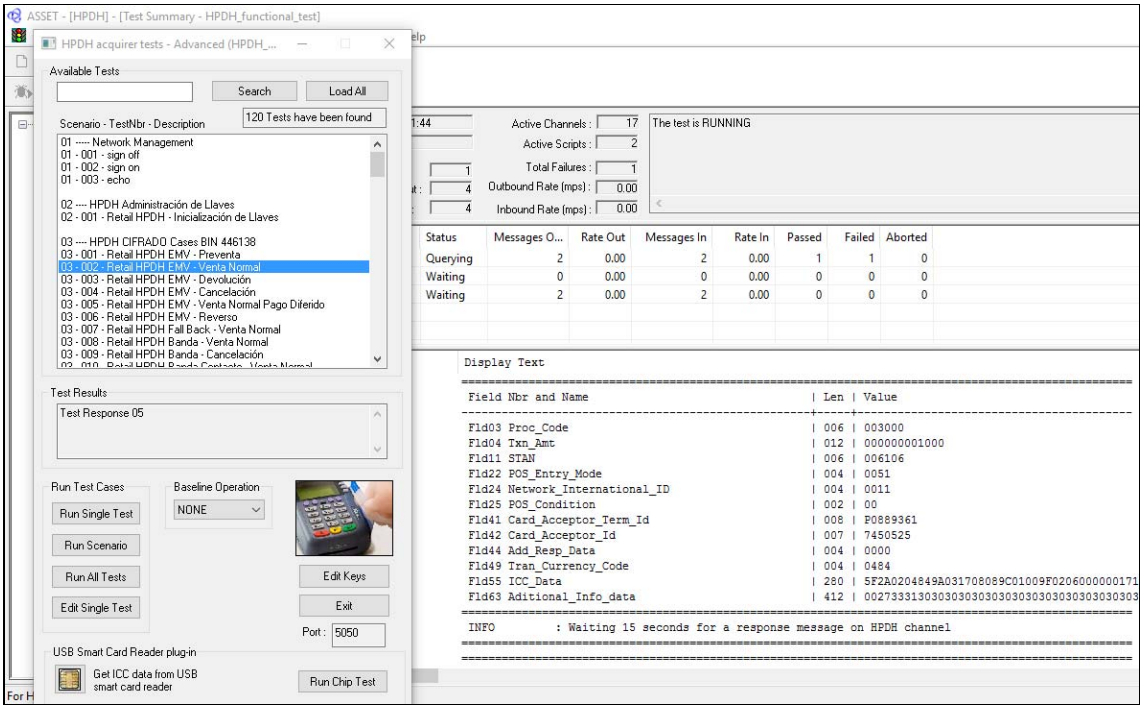

Imagen 18. Mensaje adquirente de transacción de Venta Normal con información de datos sensibles cifrada. (ACI Payment Testing, 2016)

En este caso el campo 35 (Track 2) no se encuentra presente y en su lugar viaja el Token EZ con la información de la tarjeta cifrada con la primer llave derivada de la IEK actual (B24454972D5ACFFC 02C515E219FD8864) que fue obtenida aplicando DUKPT.

En el ejemplo que se muestra el Track2 utilizado de prueba es el “5583220000000016D180120100000959” del cual aplicando el cifrado por bloques con la IEK actual se obtiene la parte del Token EZ que contienen la información del Track2 cifrado, ver información de Tokens en la imagen 19 de transacción de Venta Normal.

Tokens con información de datos sensibles cifrada

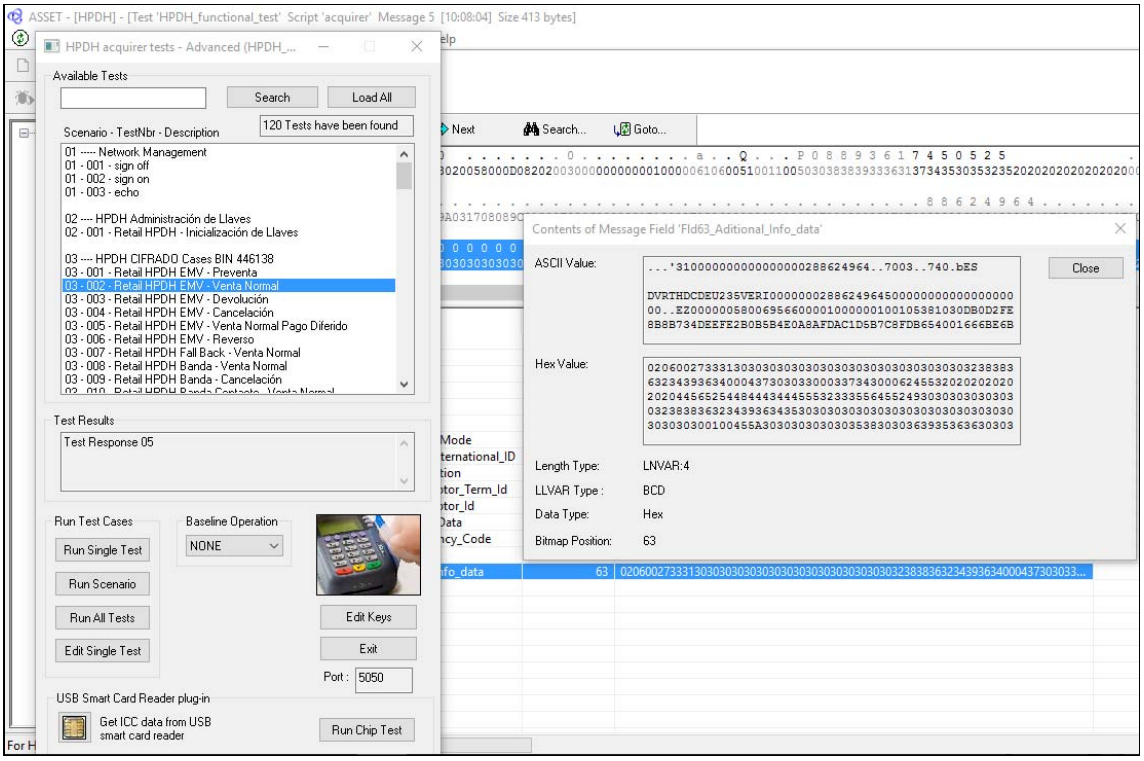

Imagen 19. Campo 63 con información de Tokens ES y EZ con información de datos sensibles cifrados. (ACI Payment Testing, 2016)

En la imagen 20, se ilustra el proceso por el cual el HSM ejecuta el comando XA (Thales HSM 8000, 2002), encargado de regenerar la llave IEK utilizando la Llave Maestra Local y KSN incluidos en el Token EZ.

Comando XA en el HSM de regenración de llave inicial IEK

```
 2483     92 17/08/08 10:07:49.65                          772935       2 -

      SOURCE P1H^RAM^10         ( PRO 67 )
      DEST   S1H^RAM10^01       ( STA 79 )

             0: 3033 3537 5841 3231 5536 3439 4142 3933 [0357XA21U649AB93]
            16: 4442 3343 4330 3539 4144 3030 3830 3341 [DB3CC059AD00803A]
            32: 4231 3930 3843 3330 3239 3036 3030 3030 [B1908C3029060000]
            48: 3030 3538 3030 3639 3536 3630 3030 3031 [0058006956600001]
            64: 3030 3138 DB0D 2FE8 B8B7 34DE EFE2 B0B5 [0018../...4.....]
            80: B4E0 A8AF DAC1 D5B7 C8FD B654           [..........T]
```

Imagen 20. Procesamiento de información en el módulo de seguridad HSM con información de información sensible cifrada con 3DES derivada por DUKPT, comando XA. (Thales HSM 8000, 2002).

Como respuesta el HSM entrega el comando XB (Thales HSM 8000, 2002), con información desencriptada del Track2 que será enviada en claro a través de un canal seguro al Banco emisor de la tarjeta, ver imagen 21.

Comando XB del HSM con Track2 desencriptado

```
 2484     36 17/08/08 10:07:49.65                          772936       0 -

      SOURCE S1H^RAM10^01       ( STA 79 )
      DEST   P1H^RAM^10         ( %177777 )

             0: 3033 3537 5842 3030 3030 3138 5583 2200 [0357XB000018U.".]
            16: 0000 0016 D180 1201 0000 0959 0000 0F00 [...........Y....]
            32: 0FFF FFFF                               [....]
```

Imagen 21. Comando XB con Track2. (Thales HSM 8000, 2002).

Mensaje 0210 como respuesta al Script HPDH de la trnsacción de Venta Normal, con información del campo 63, ver imagen 21.

Respuesta de Venta Normal a Script HPDH

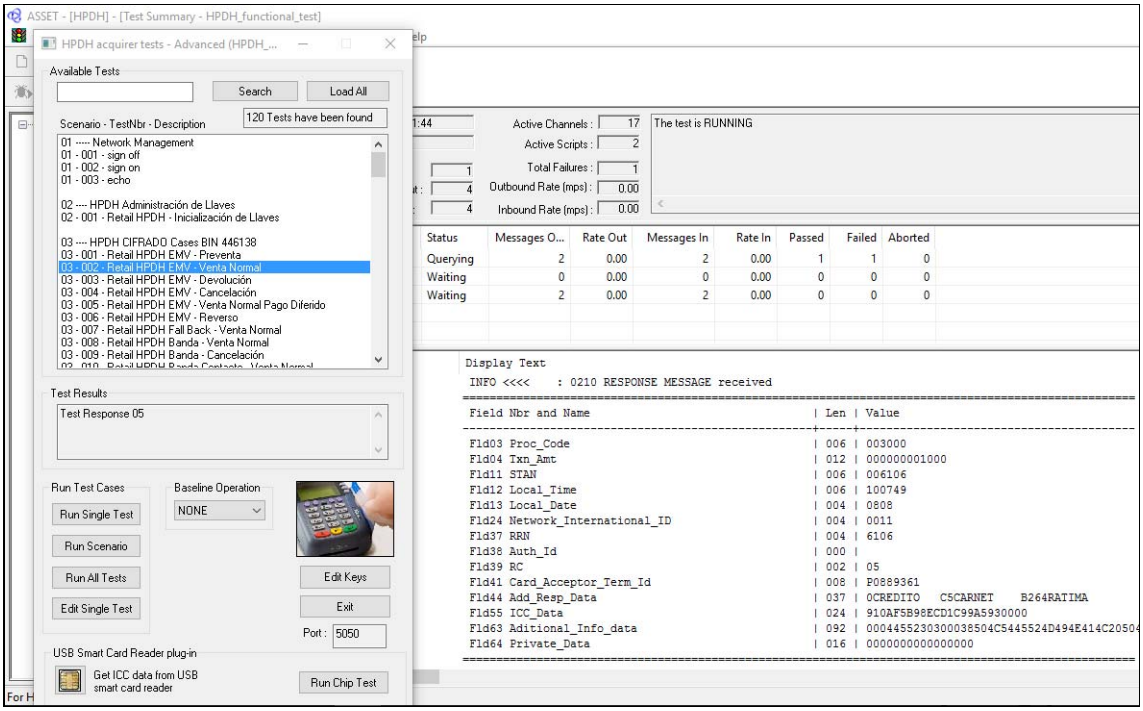

Imagen 21. Respuesta del Host Bancario al Script Simulando la Terminal Punto de Venta. . (ACI Payment Testing, 2016)

En la imagen 22 se presenta el contenido del Token de respueta ER igual a 00, indicando que el proceso de cifrado se realizó de manera exitosa por parte del script adquirente como del Host de PROSA. Adicionalmente se presenta el Token PL que indica el estado de la terminal, para efectos de la prueba la terminal no se encuentra activa.

Tokens de respuesta con información de procesamiento de DUKPT

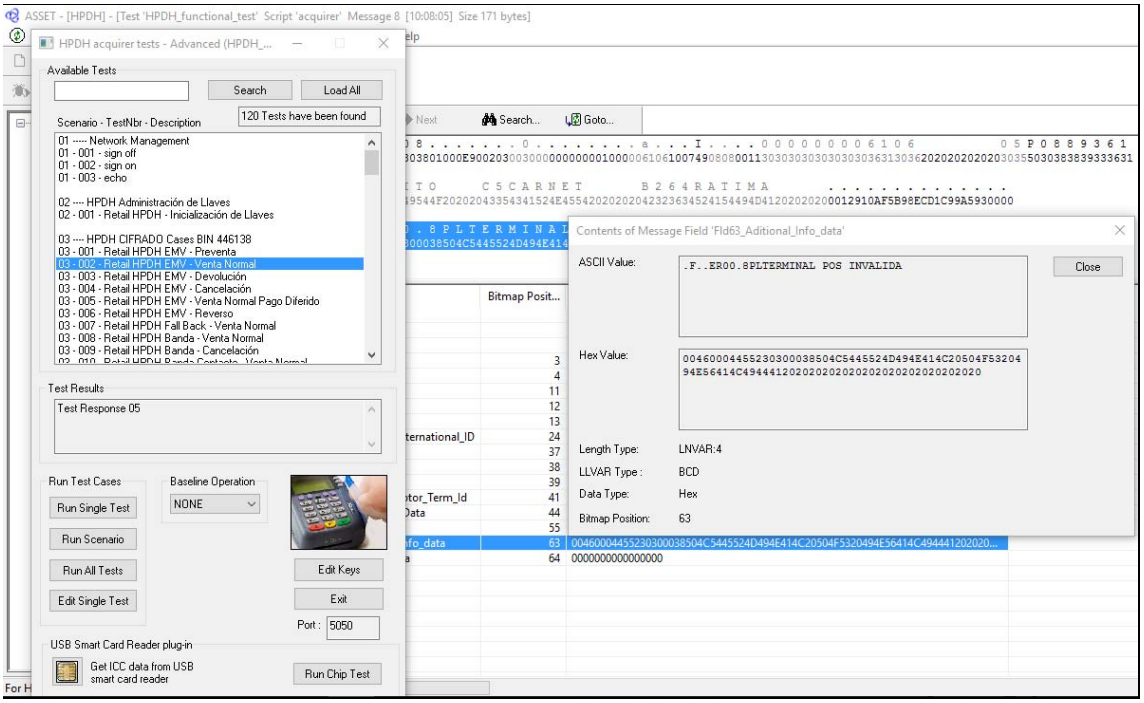

Imagen 22. Respuesta del Host Bancario al Script Simulando la Terminal Punto de Venta. . (ACI Payment Testing, 2016)

En su momento para efectos de facilitar la implantación del proyecto en producción, se definió que mientras la terminal no haya hecho su primera inicialización de llaves los datos serían entregados en claro, por lo que el script contempla en su programación el flujo de los datos en sus dos modalidades: en claro y cifrados.

Transaccionando sin Cifrado

En el caso que la Terminal o el Script no realizan el proceso de Inicialización de llaves o este tiene una respuesta de rechazo por parte de Host de PROSA en el Token ER, la terminal deberá operar sin cifrado de datos sensibles enviado los datos de la tarjeta en sus respectivos campos del protocolo HPDH o ISO8583 según corresponda, ver imagen 23.

Venta Normal sin Cifrado

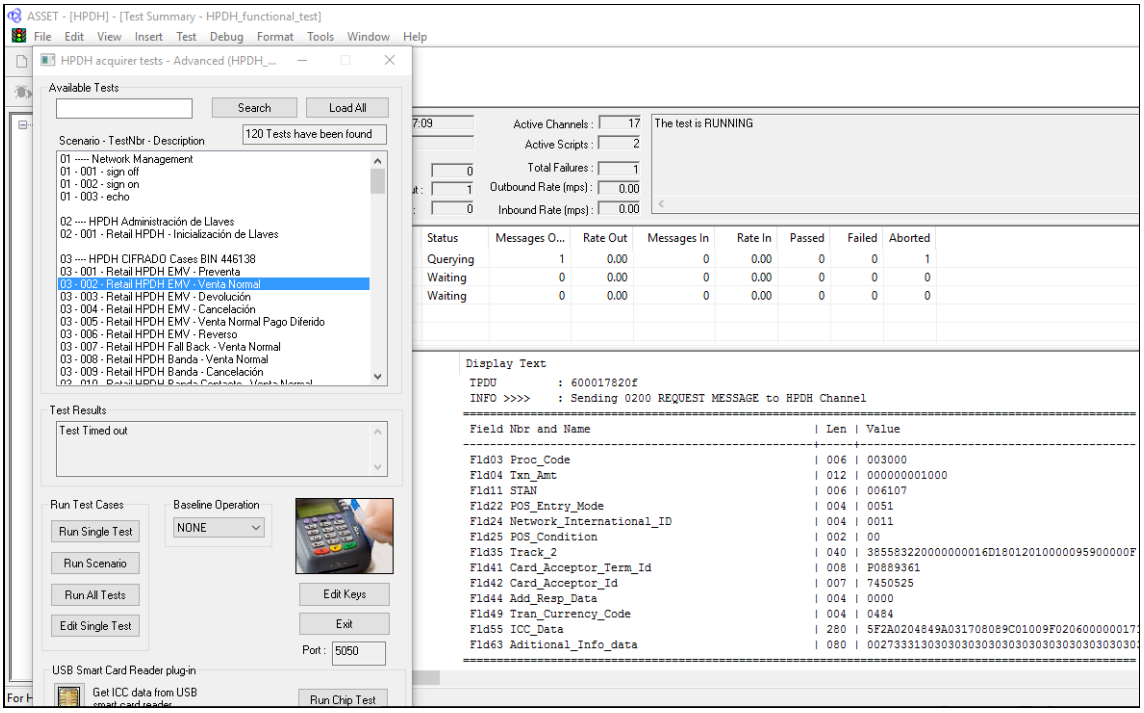

Imagen 23. Respuesta del Host Bancario al Script Simulando la Terminal Punto de Venta. . (ACI Payment Testing, 2016)

En la imagen 24 se presenta el contenido del campo 35 (Track 2) donde se puede ver que es enviado en claro, por lo que quedara a consideración del Banco Emisor de la tarjeta el declinar la transacción o no, cuando esta provenga de un comercio que debería aplicar cifrado a la transacción, como se indica en el Anexo 2 (Token ES).

Información de datos de tarjeta en claro desde el Script de Asset.

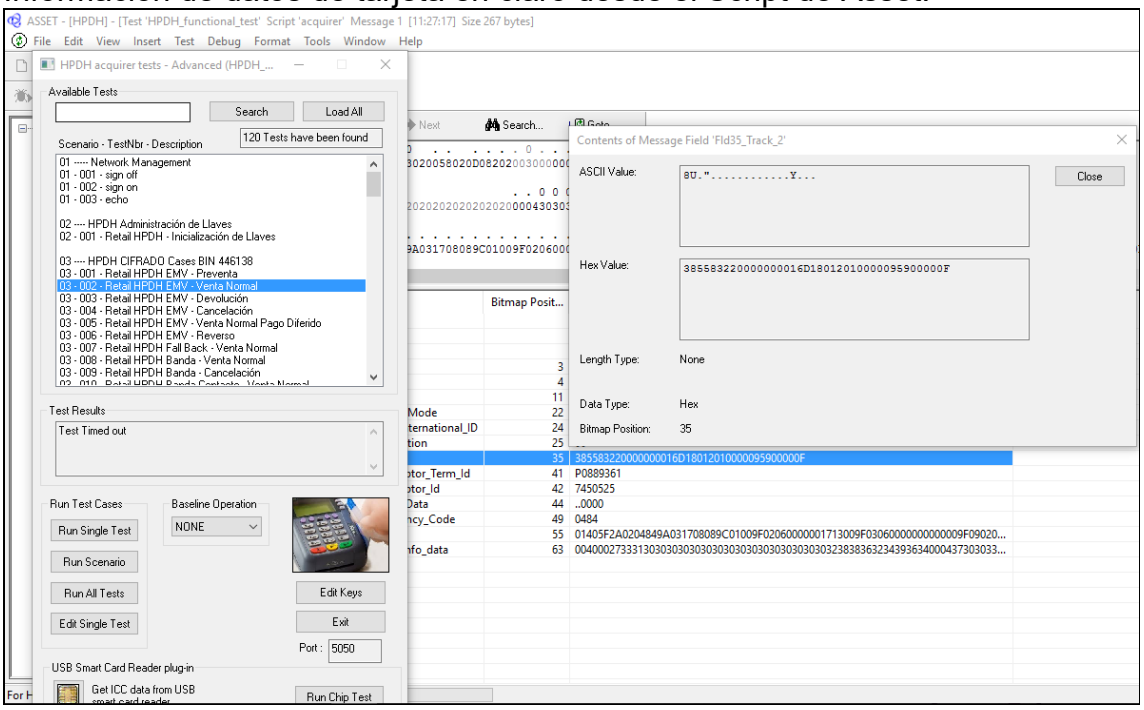

Imagen 24. Track2 en claro en el Script HPDH de ASSET. (ACI Payment Testing, 2016)

Para que el script pueda operar de manera correcta, se definió que previo al procesamiento de cualquier transacción se valide el indicador de si los datos serán cifrado o no, ya que la lógica del script deberá actuar para cifrar la información sensible ó bien mandarla en claro, pero nunca las dos modalidades al mismo tiempo.

Exclusivamente cuando la terminal este configurada para procesar con cifrado y se presente un problema en la tabla de BINES Locales, los datos sensitivos podrán ser enviados en claro y la configuración de cifrado en el Token ES viajará indicando que el Punto de Venta está configurado para cifrar, este caso no formará parte del proceso de simulación en Asset.

VII. IMPACTO DE LA EXPERIENCIA LABORAL

Como parte de la formación académica dentro de la carrera de Ingeniaría en Computación, una de las inquietudes que tenía era conocer cómo se desarrollaba el proceso de los medios de Pago Electrónico, el cual durante mi etapa de estudiante comenzaba a tener un auge mayor en el país con el uso ya no solamente en las redes de cajeros automáticos, si no como instrumento de pago de servicios a través del uso de tarjetas bancarias en diferentes canales de comunicación.

Basado a las habilidades desarrolladas en Programación y Bases de datos, inicie en la compañía participando en la automatización de reportes de la factura de Telmex a PROSA de las Terminales Punto de Venta, que se comunicaban por medio de línea telefónica a los diferentes nodos ubicados en los estados de la república, así como los que utilizaban LADA 800 como número de marcación principal o de respaldo, lo que me sirvió para comenzar a comprender la forma en que se desarrollaban los medios de pago electrónico en Terminales Punto de Venta.

Posteriormente, un paso importante con la experiencia obtenida en los primeros años de laborar en PROSA y sumado a los conocimientos adquiridos en la formación académica de Electrónica y Sistemas Digitales, fue conocer la mensajería y protocolos de comunicación utilizados por los dispositivos e Instituciones Bancarias que participan en el procesamiento de tarjetas de crédito y débito, lo cual me permitió ser considerado para trabajar en las áreas de soporte técnico del Switch principal de transacciones, resolviendo problemas en producción y a su vez colaborando en la definición y aplicación de estándares y normas nacionales e internacionales relacionados con los medios de pago electrónico.

Formar parte del área de Arquitectura de Aplicaciones en la dirección de Tecnologías de la Información de PROSA, me ha permitido plasmar los conocimientos de la formación disciplinaria y académica de la Licenciatura en Ingeniería en Computación, aunado a la experiencia laboral en el diseño avanzado de soluciones tecnológicas que la empresa ofrece a los Bancos en Instituciones financieras participantes mediante la creación de Estándares e Instrumentos basados en las normas y regulaciones de las diferentes autoridades, así como también participar en la implementación de mejoras e innovación de procesos.

El haber desarrollado un pensamiento analítico y objetivo durante la formación académica ha sido la parte fundamental para la ejecución de las tareas y funciones que hoy desempeño, así como para la implementación de la solución que se desarrolló, ya que esta permite satisfacer una importante necesidad de uso de tecnologías en las áreas de certificación de proyectos haciendo uso de la simulación.

En lo personal, el poder hacer uso los conocimientos adquiridos en la formación académica, en combinación con la experiencia en el campo laboral, representa una gran satisfacción, ya que se descubre el sentido que tiene la aplicación de estos en un campo donde los resultados se ven capitalizados de forma directa en soluciones dentro de una organización.

Algo importante en la práctica de la profesión es que las bases adquiridas en la formación académica se vuelven fundamentales cuando se aplican a tecnologías para brindar soluciones, pero esto a la vez demanda el no perder de vista el desarrollo de las nuevas que van surgiendo, lo que nos obliga a mantener los conocimientos vigentes y buscar estar siempre actualizados.

VIII. REFERENCIAS DE CONSULTA

ACI Payment Testing. (2016). *ASSET Technical Guide v5.6*. ACI Worldwide, Inc.

Reyes A. (1998). *Administración por Objetivos*. México: Limusa.

Fustér A. (1997). *Técnicas Criptográficas de protección de datos.* España: RA-MA.

ANSI X9.24-1. (2017). *Retail Financial Services Symmetric Key Management Part 1: Using Symmetric Techniques*. American National Standards Institute. Estados Unidos.

Apoyo Informática. (2017). Seguridad de la Información. Recuperado de: http://apoyoinformatica1.blogspot.mx/p/seguridad-de-la-informacion.html. Consultado el 18 de julio de 2017.

BASE24 Product Documentation. (2004). *Release 6.0 Version 5. CD-AE000-06*. ACI Worldwide, Inc.

Bitzipper (2017). *Encriptación AES - seguridad de datos*. Recuperado de: http://www.bitzipper.com/es/aes-encryption.html. Consultado el 26 de abril de 2017.

Capitulo X. (2017). *Disposiciones de Carácter General Aplicables a las Instituciones de Crédito.* México: Comisión Nacional Bancaria y de Valores CNBV.

CPSS Libro Rojo (2011). *Sistemas de pago, compensación y liquidación en México*. Banxico.

Cryptomathic (2017). *Keyshare Generator*. Recuperado de: http://extranet.cryptomathic.com/keyshares/index. Consultado el 18 de julio de 2017.

DEVNULL. (2016). *Algoritmos de Cifrado*. Recuperado de: https://elbinario.net/2016/04/05/algoritmos-de-cifrado-i/. Consultado el 08 de agosto de 2017.

Tarifa E. (2015). *Teoría de Modelos y Simulación*. Argentina: Facultad de Ingeniería - Universidad Nacional de Jujuy.

Espíndola J.L. (1996). *Análisis de problemas y toma de decisiones*. México: Logman de México Editores.

First Data (2002). *Technical documentation FDR Bank Card*. USA: First Data Resources Inc.

Fishman G.S. (1978). *Conceptos y métodos en la simulación digital de eventos discretos*. México: Limusa.

Gesfor México. (2017). *Core Bancario*. Recuperado de: http://www.gesfor.com.mx/core-bancario. Consultado el 24 de marzo de 2017.

ICSF. (2014). *Cryptographic Services ICSF Application Programmer's Guide SA22-7522-16*. Recuperado de: https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.csfb400/trame1.htm. Consultado el 26 de abril de 2017.

Information Technology Infrastructure Library (2017). *Manual de ITIL v3*. Recuperado de: http://www.knowledgetransfer.net/dictionary/ITIL/en/. Consultado el 05 de abril de 2017.

Instinto Lócgico (2017). *Cifrar es necesario*. Recuperado de: http://instintologico.com/introduccion-a-la-criptografia-es-necesario-cifrar-criptografia-simetrica-y-asimetrica/

Introducción a las Finanzas. (2016). *¿Cómo funciona el sistema de compensación y liquidación?* Recuperado de: http://ciberconta.unizar.es/bolsa/52.htm. Consultado el 18 de diciembre de 2016.

MasterCard International Incorporated. (2001). *Customer Interface Specification (CIS)*. USA: Global Member Operations Support.

MasterCard International Incorporated. (2000). *MasterCard Debit Switch Online Specifications*. USA: Global Member Operations Support.

MasterCard International Incorporated. (1987). *ISO 8583–1987 Data Element Definitions*.

Omnipays. (2016). *Switches Transaccionales Bancarios: Un ejemplo de operatividad eficiente*. Optima Technology S.A. de C.V. Recuperado de: http://omnipays.com/index.php/articulos/85-switches-transaccionales-bancarios. Consultado el 08 de agosto de 2017.

PKCS #1 V2.2 (2012). *RSA Cryptography Standard*. RSA Laboratories

Paypal. (2017). *3D-Secure, sistema de pagos seguros*. Recuperado de: https://www.paypal.com/es/webapps/mpp/3dsecure-faqs. Consultado el 02 de abril de 2017.

Revista Retailling. (2017). *Retailers*. Recuperado de: http://www.revistaretailing.org/desarrollo_noticia.php?id_noticia=32. Consultado el 07 de enero de 2017.

Zorrilla S. (1986). *Introducción a la Metodología de la Investigación. Casos Aplicados a la Administración*. México: Océano.

Shannon R.E. (1988). *Simulación de Sistemas. Diseño, desarrollo e implementación.* México: Trillas.

Tecnocomputación 3000. (2017). *Pin Pads*. Recuperado de: http://www.tecnocomputacion.com/soluciones/puntos-de-venta/pin-pads/. Consultado el 15 de abril de 2017.

Techopedia. *Electronic Cash Registers*. (2017). Recuperado de: https://www.techopedia.com/definition/26650/electronic-cash-register-ecr Consultado el 18 de julio de 2017.

Thales HSM 8000. (2002). *HSM 8000 Host Command Reference Manual*. USA: THALES e-SECURITY LTD.

Visa International. (2000). *V.I.P. System BASE I Technical Specifications*. USA: VisaNet Business Enhancements.

Visa International. (2009). *V.I.P. System SMS ATM Processing*. USA: VisaNet Business Enhancements.

IX. ANEXOS

ANEXO 1. Esquema transaccional Emisor-Adquirente a ser implementado como parte de la simulación de mensajes financieros bajo DUKPT.

Esquemas adquirentes soportados.

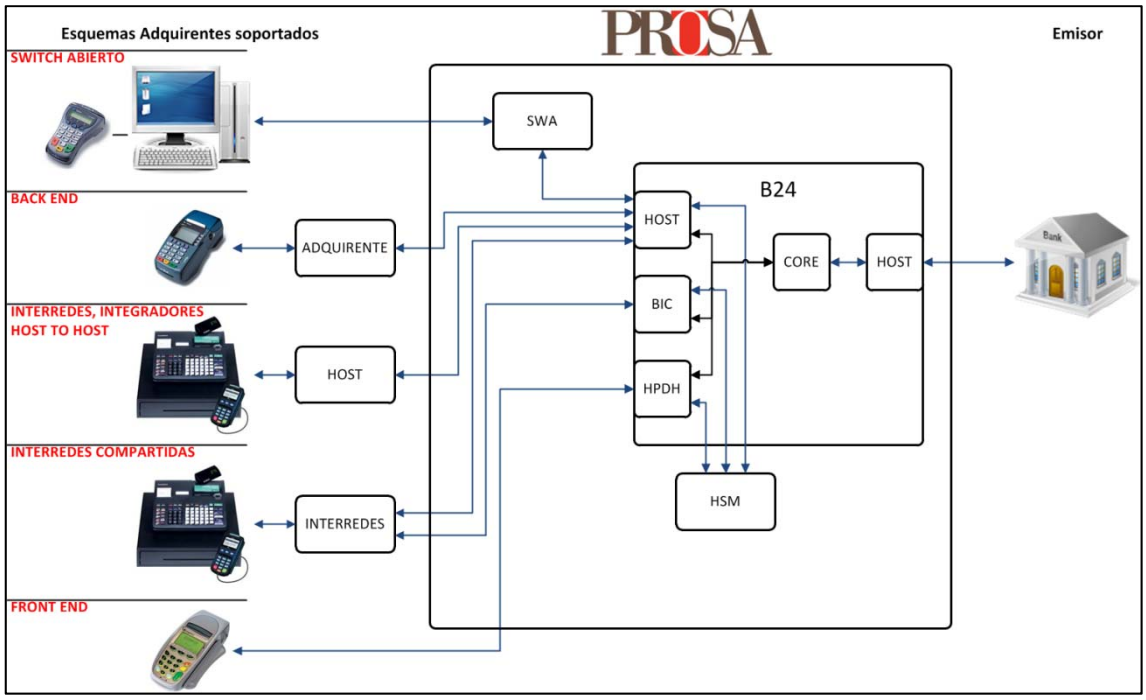

Imagen 25. Diagrama conceptual de operación Emisor - Adquirente. (CNVB Capítulo X, 2010)

De acuerdo a lo estipulado por la CNVB en el Capítulo X, todos los adquirentes que realicen procesamiento deberán asegurar el envio de la información cifrada desde el punto a recepción de la tarjeta hasta la entrega al Host Bancario correspondiente.

ANEXO 2. Descripción de Tokens para Transacciones Financieras.

La mayor parte del contenido de estos Tokens debe ser tomado desde el Dispositivo Punto de Venta, excepto cuando la descripción del campo indique lo contrario.

Token EW. *Requerimiento de Generación de Nueva Llave*

#	Nombre	Inicio	Fin	Long	Formato	Valores válidos
H-1	EYE-CATCHER	1	1	1	X(01)	Header de Token: Identificador de inicio de token. **"!"** valor fijo
H-2	USER-FLD1	2	2	1	X(01)	Header de Token: Separador 1. **" "** valor fijo
H-3	Identificador del Token	3	4	2	X(02)	Header de Token: Identificación del token que se está enviando. **"EW"** valor fijo
H-4	Longitud de datos	5	9	5	9(05)	Header de Token: Longitud de la sección de datos del token. **00538** valor fijo
H-5	USER-FLD2	10	10	1	X(01)	Header de Token: Separador 2. **" "** valor fijo
1	Llave aleatoria cifrada	11	522	512	X(512)	Contiene la llave aleatoria de doble longitud cifrada con la llave pública RSA.
2	Check Value	523	528	6	9(06)	Contiene el resultado de validación de la llave aleatoria de doble longitud.
3	Versión Llave RSA Publica	529	538	10	X(10)	Contiene el Nombre / Versión de la Llave RSA Pública con que el PIN PAD ha cifrado la llave aleatoria del Subcampo 1 de este mismo token.
4	Algoritmo de Padding	539	540	2	X(02)	Contiene el Método de Padding con el que la Llave Aleatoria fue cifrada. 01 – PKCS 1.5 02 – OAEP (Uso futuro)
5	CRC32 de Llave Cifrada	541	548	8	X(08)	Valor de verificación generado por el PIN PAD, usando algoritmo CRC32, sobre Subcampo 1 de este mismo Token.

Token EX. *Respuesta de Generación de Nueva Llave*

#	Nombre	Inicio	Fin	Long	Formato	Valores válidos
H-1	EYE-CATCHER	1	1	1	X(01)	Header de Token: Identificador de inicio de token. **"!"** valor fijo
H-2	USER-FLD1	2	2	1	X(01)	Header de Token: Separador 1. **" "** valor fijo
H-3	Identificador del Token	3	4	2	X(02)	Header de Token: Identificación del token que se está enviando. **"EX"** valor fijo

#	Nombre	Inicio	Fin	Long	Formato	Valores válidos
H-4	Longitud de datos	5	9	5	9(05)	Header de Token: Longitud de la sección de datos del token. **00068** valor fijo
H-5	USER-FLD2	10	10	1	X(01)	Header de Token: Separador 2. **“ “** valor fijo
1	Llave nueva cifrada	11	42	32	X(32)	Contiene la nueva llave cifrada con la llave aleatoria original. **NOTA:** Puede viajar en ESPACIOS si el campo 4 es diferente a “00”
2	KSN inicial	43	62	20	X(20)	Ej.: “0102012345678AE00001” **NOTA:** Puede viajar en ESPACIOS si el campo 4 es diferente a “00”
3	Check Value Nueva Llave Inicial	63	68	6	X(06)	Contiene el Check Value de la nueva Llave inicial. **NOTA:** Puede viajar en ESPACIOS si el campo 4 es diferente a “00”
4	Estatus de la solicitud de Nueva Llave	69	70	2	9(2)	Código de resultado. 00 – Nueva llave correctamente generada. 01 – Check Value de llave de Transporte Incorrecto. 02 – Llave RSA Inexistente. 03 – Bloque cifrado RSA corrupto (no concuerda CRC). 04 – Problemas con HSM (módulo encriptor).
5	CRC32 de nueva Llave Cifrada	71	78	8	X(08)	Valor de verificación generado por el Host Bancario, usando algoritmo CRC32, sobre Subcampo 1 de este mismo Token. **NOTA:** Puede viajar en ESPACIOS si el campo 4 es diferente a “00”

Token ER. *Resultado de Cifrado*

#	Nombre	Inicio	Fin	Long	Formato	Valores válidos
H-1	EYE-CATCHER	1	1	1	X(01)	Header de Token: Identificador de inicio de token. ¡ Valor fijo.
H-2	USER-FLD1	2	2	1	X(01)	Header de Token: Separador 1. **“ “** Valor fijo.
H-3	Identificador del Token	3	4	2	X(02)	Header de Token: Identificación del token que se está enviando. **ER** Valor fijo.
H-4	Longitud de datos	5	9	5	9(05)	Header de Token: Longitud de la sección de datos del token. **00002** Valor fijo.
H-5	USER-FLD2	10	10	1	X(01)	Header de Token: Separador 2. **“ “** Valor fijo.

#	Nombre	Inicio	Fin	Long	Formato	Valores válidos
1	Bandera Actualización Llaves	11	11	1	X(1)	En requerimiento vendrá en ESPACIOS. En Respuestas: **0** – No inicializar llaves **1 –** Sugerir al operador de la caja que debe Inicializar Llaves → **programar que la Caja realice una inicialización de Llaves al apagar y volverla a encender.** **2 –**Inicialización de Llaves Obligatoria → **La caja debe realizar una inicialización de llaves al terminar la transacción actual.**
2	Bandera Actualización BINES	12	12	1	X(1)	En requerimiento vendrá en ESPACIOS. En Respuestas: **0** – No actualizar BINES **1 –** Actualizar BINES (usar Token ET que se adjunta en la misma respuesta).

Token ES. *Terminal Status – Configuración Cifrado*

#	Nombre	Inicio	Fin	Long	Formato	Valores válidos
H-1	EYE-CATCHER	1	1	1	X(01)	Header de Token: Identificador de inicio de token. ¡ Valor fijo.
H-2	USER-FLD1	2	2	1	X(01)	Header de Token: Separador 1. **“ “** Valor fijo.
H-3	Identificador del Token	3	4	2	X(02)	Header de Token: Identificación del token que se está enviando. **ES** Valor fijo.
H-4	Longitud de datos	5	9	5	9(05)	Header de Token: Longitud de la sección de datos del token. **00060** Valor fijo.
H-5	USER-FLD2	10	10	1	X(01)	Header de Token: Separador 2. **“ “** Valor fijo.
1	Versión Software	11	30	20	X(20)	Nombre / Versión de la aplicación en el PIN PAD
2	Serie del PIN PAD	31	50	20	X(20)	Marca y Número de Serie del Dispositivo
3	Configuración de Cifrado	51	51	1	X(1)	Indica si el PIN PAD tiene activa la capacidad de cifrado. 0 – Configurado para NO cifrar. 5 – Activo para Cifrado de Datos Sensitivos con DUKPT

#	Nombre	Inicio	Fin	Long	Formato	Valores válidos
4	ID Tabla de BINES Locales informado por la Caja	52	59	8	X(08)	IDENTIFICADOR alfanumérico asignado al comercio, para asociar en el HOST Bancario una tabla de BINES locales que se deberá transmitir al PIN PAD. La caja deberá entregar este identificador al PIN PAD, para su registro interno y posterior reenvío al Host.
5	ID Tabla de BINES Locales cargado en PIN PAD	60	67	8	X(08)	IDENTIFICADOR alfanumérico de la tabla de BINES locales que esté cargada en el PIN PAD. **NOTA:** Este dato deberá ser el retornado por el PIN PAD, será el valor “00000000” si el PIN PAD nunca ha cargado una tabla de BINES Locales.
6	Versión Tabla de BINES Locales cargada en PIN PAD	68	69	2	X(2)	Valor numérico para identificar la versión de la tabla de BINES locales. Si el PIN PAD no tiene una tabla de BINES cargada, debe enviar “00”. Valores válidos del “00” al “FF”.
7	Bandera Petición de Nueva Llave	70	70	1	X(1)	Indica si el PIN PAD está pidiendo inicialización de Llaves. “1” – Se requiere nueva llave. Token EW debería estar presente. “0” u otro valor – No se pide nueva llave.

Token ET. *Tabla de BINES que no Cifran*

#	Nombre	Inicio	Fin	Long	Formato	Valores válidos
H-1	EYE-CATCHER	1	1	1	X(01)	Header de Token: Identificador de inicio de token. ¡ Valor fijo.
H-2	USER-FLD1	2	2	1	X(01)	Header de Token: Separador 1. **“ “** Valor fijo.
H-3	Identificador del Token	3	4	2	X(02)	Header de Token: Identificación del token que se está enviando. **ET** Valor fijo.
H-4	Longitud de datos	5	9	5	9(05)	Header de Token: Longitud de la sección de datos del token. **00366** Valor fijo.
H-5	USER-FLD2	10	10	1	X(01)	Header de Token: Separador 2. **“ “** Valor fijo.
1	ID Tabla de BINES Locales	11	18	8	X(08)	ID de la tabla de BINES contenida en este Token, valor que debería corresponder al mismo entregado por el comercio en el Subcampo 4 del Token ES.

#	Nombre	Inicio	Fin	Long	Formato	Valores válidos
2	Versión de la Tabla de BINES Locales	19	20	2	X(02)	Versión de la tabla de BINES Locales que se está actualizando en este Token al dispositivo. Valores válidos del “01” al “FF”.
2	KSN para Cifrado de Tabla	21	40	20	X(20)	Key Serial Number con el cual se ha diversificado la llave BDK en el Host Bancario para cifrar la Tabla de BINES
3	Longitud Usada del BUFFER CIFRADO	41	44	4	9(4)	Cantidad real de posiciones usadas en el Sumcampo 5, en múltiplos de 16. Valor Máximo: 320 (20 bloques)
4	Longitud Real Tabla de la Tabla de BINES Locales	45	48	4	9(4)	Corresponde a la cantidad real de posiciones que ocupa la tabla de BINES Locales una vez que el PIN PAD ha descifrado el BUFFER. Valor Máximo: 320
5	BUFFER CIFRADO con Tabla de BINES Locales	49	368	320	X(320)	BLOQUES de la tabla de BINES CIFRADOS usando 3DES inverso. Las posiciones no usadas deberán viajar con letras “F”.
6	CRC32 de BUFFER CIFRADO	369	376	8	X(08)	CRC32 calculado sobre la porción usada del *BUFFER CIFRADO* contenido en el subcampo anterior. Esta porción está definida por la longitud indicada por el Subcampo 3 de este mismo Token. Valores permitidos del “0” a la “F”.

Token EY. *Cifrado del Track1*

#	Nombre	Inicio	Fin	Long	Formato	Valores válidos
H-1	EYE-CATCHER	1	1	1	X(01)	Header de Token: Identificador de inicio de token. **“!”** valor fijo.
H-2	USER-FLD1	2	2	1	X(01)	Header de Token: Separador 1. **“ “** valor fijo.
H-3	Identificador del Token	3	4	2	X(02)	Header de Token: Identificación del token que se está enviando. **“EY”** valor fijo
H-4	Longitud de datos	5	9	5	9(05)	Header de Token: Longitud de la sección de datos del token. **00172** valor fijo
H-5	USER-FLD2	10	10	1	X(01)	Header de Token: Separador 2. **“ “** valor fijo
1	Longitud del TRACK1	11	14	4	9(04)	EJ: “0076” Identifica la longitud del track1 en claro, antes del proceso de cifrado (longitud original, no multiplicada x 2), si esta viene en “0000” no hay bandera de track1 encendida. Valor máximo: 80.

#	Nombre	Inicio	Fin	Long	Formato	Valores válidos
2	Datos de track1 Cifrados	15	174	160	X(160)	Contiene el TRACK1 cifrado.
3	CRC32 de Track1 Cifrado	175	182	8	X(08)	Valor de verificación generado por el PIN PAD, usando algoritmo CRC32, sobre Subcampo 2 de este mismo Token. Valores del “0” a la “F”.

Token EZ. *Banderas y Datos Sensitivos Cifrados*

#	Nombre	Inicio	Fin	Long	Formato	Valores válidos
H-1	EYE-CATCHER	1	1	1	X(01)	Header de Token: Identificador de inicio de token. **“!”** valor fijo
H-2	USER-FLD1	2	2	1	X(01)	Header de Token: Separador 1. **“ “** valor fijo
H-3	Identificador del Token	3	4	2	X(02)	Header de Token: Identificación del token que se está enviando. **“EZ”** valor fijo
H-4	Longitud de datos	5	9	5	9(05)	Header de Token: Longitud de la sección de datos del token. **00098** valor fijo
H-5	USER-FLD2	10	10	1	X(01)	Header de Token: Separador 2. **“ “** valor fijo
1	Key Serial Number	11	30	20	X(20)	Ej.: “01020123456789E00001”
2	Contador Real de Cifrados	31	37	7	9(07)	EJ: “0999999” Identifica cuántos cifrados ha hecho un PIN PAD desde la última carga de llave. Valor máximo es de “1000000” (1 millón).
3	Contador de Cifrados Fallidos Consecutivos	38	39	2	9(02	EJ: “00 Identifica cuántas transacciones le han declinado los Hosts Bancarios consecutivamente a la CAJA por falla en el cifrado (respuesta en ISO39 = 70). Este campo es importante para determinar si se debe aplicar una INICIALIZACION DE LLAVES. Debe ser actualizado y entregado por la CAJA. El valor debe ser regresado a “00” por la CAJA cuando reciba la primera respuesta con ISO39 <> “70” desde cualquier Host Bancario.
4	Bandera de TRACK2	40	40	1	X(01)	EJ: ‘0’ – No hay track2 ‘1’ – Hay track2 Indica la presencia del TRACK2
5	Modo de Lectura de la Tarjeta	41	42	2	X(02)	01 – Tarjeta Digitada 90 – Lectura de Banda. 80 – Fallback 05 – Lectura del CHIP.

#	Nombre	Inicio	Fin	Long	Formato	Valores válidos
6	Longitud de TRACK2	43	44	2	9(02)	EJ: “37” Identifica la longitud del track2 en claro, si esta viene en “00” no hay bandera de track2 encendida.
7	Bandera de CVV2	45	45	1	X(01)	EJ: ‘0’ – no hay cvv2 ingresado ‘1’ – se ingresó cvv2 ‘A’ – El PIN PAD no solicitó el CVV2 Indica si hubo captura del código de seguridad.
8	Longitud de cvv2 en claro	46	47	2	9(02)	EJ: “03”, máximo ‘05’ Indica la longitud del cvv2 en claro.
9	Bandera de TRACK1	48	48	1	X(01)	EJ: ‘0’ – No hay track1 ‘1’ – Hay track1 (Nota: el track1 solo está presente en transacciones con banda) Indica la presencia del TRACK1
10	Datos Sensitivos Cifrados	49	96	48	X(48)	Contiene el TRACK2 y el código de seguridad (CVV2) cifrado.
11	4 últimos Dígitos del PAN	97	100	4	X(04)	4 últimos dígitos en claro del PAN de la tarjeta que la caja recibió en claro desde el PIN PAD. Este valor debe entregado POR LA CAJA.
12	CRC32 sobre Datos Sensitivos	101	108	8	X(8)	CRC32 sobre el Subcampo 10, valor calculado por el PIN PAD. Valores del “0” a la “F”.

ANEXO 3. Códigos de Resultado ISO (Campo 39)

CODIGO ISO	SIGNIFICADO ISO	SIGNIFICADO ADQUIRENTES	Causas Posibles	MENSAJE COMERCIO
70	**Reserved for ISO use**	**Error descifrando Track2**	PROSA al descifrar NO llega a los 4 últimos dígitos de la tarjeta enviados por la CAJA. o Puede que la terminal haya perdido la llave. o Puede ser un BUG en la terminal que informa un nombre de llave diferente.	**Reintente**
71	**Reserved for ISO use**	**Debe inicializar llaves**	PROSA determina que la Terminal debe cargar una llave por primera vez o renovar la llave cargada. o El comercio ya opera con cifrado y la Terminal es nueva, nunca ha inicializado llaves. o La Terminal ha excedido un umbral de errores del tipo 70 (Error descifrando Track2) o La llave en la Terminal ha excedido el millón de cifrados y ya no puede continuar cifrando.	**Debe inicializar la Terminal**
72	**Reserved for ISO use**	**Problema inicializando Llaves**	Este error sólo aplica para la Inicialización de Llaves. o Nombre de la Llave RSA Pública es erróneo o no existe en PROSA → Bug en Terminal o Check Value de la llave TK Aleatoria descifrada en PROSA no concuerda con la notificada → Bug en Terminal al transmitir o Problemas en PROSA → HSM retorna error	**Reintente – Llamar a Soporte**
73	**Reserved for ISO use**	**Error en CRC**	La CAJA o aplicación del comercio alteraron o truncaron los datos cifrados entregados por la terminal o Error que se puede dar si el comercio tiene un BUG ó ha instalado cambios sin certificar que implica que los datos cifrados NO llegan íntegros a PROSA. o Puede ser en Ventas ó en Inicialización de Llaves	**Error en el Sistema – Llamar a Soporte**

Para el caso de los códigos de rechazo 72 y 73, PROSA responderá en el campo ISO39 del mensaje financiero tipo 0210 de acuerdo a lo siguiente:

72 – “Error Inicializando Llaves”. En este caso se deberá validar el subcampo 4 del Token EX para conocer el motivo exacto del problema.

73 – “Error en CRC”. Se valida el CRC de la llave aleatoria cifrada retornada por la terminal, si esta NO es correctamente transmitida hasta el Host Bancario causa que el CRC no sea correctamente validado.

ANEXO 4.- Definición técnica del método DUKPT en el Dispositivo Punto de Venta y el Host bancario.

Algoritmo de DUKPT en el Dispositivo Punto de Venta y ASSET

“Load initial key” (Comando externo)

1. Guardar Key_register, recibido en el comando_externo de inicialización en el registro #21 de Future_Key.
2. Generar y guardar el LRC para el registro #21 de Future_Key.
3. Escribir la dirección de memoria del registro #21 de Future_Key en el Current_pointer.
4. Guardar el KSN, este es recibido en la inicialización del comando externo.
5. Clarear el Encryption_counter.
6. Setear el bit #1 (el bit de más a la izquierda) en 1 del Shift_register, seteando todos los demás bits a cero.
7. Ir a “New_key_3”

“New_key” (Etiqueta local)

1. Contar el número de bits en 1 del Encryption_counter. Si este es menor a 10 entonces ir a “New_key_1”.
2. Borrar Future_key en [Current_pointer].
3. Setear el LRC para Future_key en [Current_pointer] a un valor inválido (p.e. agregar 1 al LRC).
4. Agregar el Shift_register al Encryption_counter. (Este procedimiento causa el salto para aquellos contadores que tienen más de 10 bits encendidos).
5. Ir a “New_Key_2”.

“New_key_1” (Etiqueta local)

1. Shiftea el Shift_register un bit a la derecha
2. Si el Shift_register contiene todos los bits en cero ir a “New_key_4” si no ir a “New_key_3”

“New_key_3” (Etiqueta local)

1. El Shift_register, justificado a 64 bits, padeado a la izquierda con ceros, hacer un OR con los 64 bits de más a la derecha del KSN, transferir a Crypto_Register_1
2. Copiar Key_left_half y Key_right_half de Future_key[Current_pointer] en Key_register
3. Llamar la subrutina Non_reversible_key_generation ( ).
4. Guardar el contenido de Crypto_Register_1 en Key_left_half de Future_key indicado por la posición del bit encendido del Shift_register.
5. Guardar el contenido de Crypto_Register_2 en Key_right_half de Future_key indicado por la posición del bit encendido del Shift_register.
6. Generar y guardar el LRC para Future_key indicado por el bit encendido del Shift_Register.
7. Ir a “New_Key_1”

“New_key_4” (Etiqueta local)

1. Borrar Future_key [Current_pointer]
2. Setear el LRC para Future_key[Current_pointer] en valor inválido (incrementar el LRC en 1).
3. Sumar uno al Encryption_counter.
4. Ir a “New_key_2”.

“New_key_2” (Etiqueta local)

1. Si el Encryption_counter contiene todos los bits en cero, termina la operación (El dispositivo ahora es inoperativo ha encriptado más de 1 millón de veces). Si no son ceros entonces ir a “Exit”.

“Exit” (Etiqueta local)

1. Retorno a la llamada de la rutina original.

“Request Data” (comando externo). Aquí se cifran los datos Sensitivos y sustituye al “Request PIN Entry” del estándar

1. Insertar o deslizar la tarjeta en el dispositivo.
2. Llamar a subrutina “Get_Data”
3. Ir a “Request_Data_1”.

“Request_Data_1” (Etiqueta local). Sustituye al “Request PIN Entry 1”

1. Llamar la subrutina Set_bit ( ).
2. Escribir en [Current_pointer] la dirección del registro de Future_key indicado por la posición del bit encendido en el Shift_register.
3. Validar el LRC en Future_key[Current_pointer]. Si el LRC es correcto (Llave válida). Ir a “Request_Data_2”.
4. Si el LRC en Future_key[Current_pointer] no es correcto sumar el Shift_register al Encryption_counter (para saltar otra llave inválida).
5. Si el Encryption_counter es igual a cero, se detiene la operación. (El dispositivo de entrada es ahora inoperativo, se ha encriptado más de 1 millón de veces).
6. Ir a “Request_Data_1”.

“Request_Data_2” (Etiqueta local) Sustituye al “Request PIN Entry 2”

1. Copiar Key_left_half y Key_right_half de Future_key[Current_pointer] en Key_register.
2. En el algoritmo estándar genera una llave MAC, este paso no será utilizado por nosotros.
3. XOR con Key_register y el hexadecimal “0000 0000 00FF 0000 0000 0000 00FF 0000” (0000 0000 0000 00FF 0000 0000 0000 00FF). Esto producirá una variante de la llave para cifrar datos, almacenar en Key_register. A la llave resultante hay que aplicarle Triple_DES usándose así misma como llave y guardarla en Key_register.
4. Realizar Triple_DES de B1 .. B3 usando Key_register almacenar resultado en T1..T13.
5. Agregar 1 al Cipher_counter
6. Formatear y guardar los bloques de T1 a T13 encriptados en el mensaje, el cual incluye:
    a. Los datos del KSN con las F’s hexadecimal suprimidas. (Incluye los 21 bits del contador de encripción).
    b. El bloque de T1 y T2 encriptados.
    c. El Cipher_counter.
7. Ir a “New_key”.

Llave usada para	**Variant constant**
PIN	00000000000000FF00000000000000FF
MAC, request or both ways (MDKI)	000000000000FF00000000000000FF00
MAC, response (MDKO)	00000000FF00000000000000FF000000
Data encryption	0000000000FF00000000000000FF0000

“Power On Reset” (Comando Externo)

1. Setear a uno aquellos bits del Encryption_counter que correspondan a los registros perdidos de Future_key debido a una interrupción.
2. Incrementar el Encryption_counter.
3. Ir a “Exit”.

“Set Bit” (Subrutina local)

1. Setear en 1 el bit del Shift_register que corresponda al bit más a la derecha del Encryption_counter, haciendo todos los demás bits en el Shift_register igual a cero. Por ejemplo:
    a. Si el Encryption_counter = 0 0010 1100 1101 1100 0011, El Shift_register será = 0 0000 0000 0000 0000 0001.
    b. Si el Encryption_counter = 0 0010 1100 1101 1100 0000, El Shift_register será = 0 0000 0000 0000 01000 0000.
2. Retornar a la subrutina que hizo la llamada.

“Non reversible key generation” (Subrutina Local)

1. Crypto_Register_1 XOR con Key_register_right de Key_register almacenar en Crypto_Register_2.
2. Crypto_Register_2 DES con Key_register_left de Key_register almacenar en Crypto_Register_2.
3. Crypto_Register_2 XOR Key_register_right de Key_register almacenar en Crypto_Register_2.
4. Hacer XOR de Key_register con el hexadecimal C0C0C0C000000000C0C0C0C000000000 para generar una variante de la llave, almacenar en Key_register.

5. Crypto_Register_1 XOR con Key_register_right de Key_register almacenar en Crypto_Register_1.
6. Crypto_Register_1 DES con Key_register_left de Key_register almacenar en Crypto_Register_1.
7. Crypto_Register_1 XOR con Key_register_right de Key_register almacenar en Crypto_Register_1.
8. Regresar a la subrutina que hizo la llamada.

“Get_Data” (Subrutina local)

1. Obtener el Track1, TrackII y Código de Seguridad.
2. Construir Bloques B1 a B13 de acuerdo a definición de Bloques.
3. Retornar a la subrutina que hizo la llamada.

Algoritmo de DUKPT en el HOST

Áreas de almacenamiento

BDK → 16 bytes /* Llave de doble longitud, sirve para generar la llave con la cual se inicializará el dispositivo */
R8 → 8 bytes
R8A → 8 bytes
R8B → 8 bytes
R3 → 21 bits
SR → 21 bits similar al Shift_register

KSNR → 8 bytes; los 8 bytes de más a la derecha del Key_serial_number (KSN) recibidos desde el PIN PAD.

Ikey → 16 bytes; la llave inicialmente cargada en el PIN PAD
Ikey_new → 16 bytes; la nueva llave inicial para ser cargada en el PIN PAD

Curkey → 16 bytes; Contiene la llave de encripción de la transacción actual.

Ca → 8 bytes.

Cálculo de la llave de encripción inicial

- Ca = Los 8 bytes de más a la izquierda del KSN
- Operación AND con E0 hexadecimal y el byte 8 de Ca para obtener el número de serie

Byte 8 de Ca **&** E0x → Ca = número de serie inicial

- Realizar Triple DES de Ca con la llave inicial y guardar en la mitad izquierda de Ikey.
- Realizar operación XOR con la BDK y el hexadecimal C0C0C0C000000000C0C0C0C000000000
- Realizar TRIPLE DES de CA con la variación de la llave inicial y guardar el resultado en la mitad derecha de Ikey.
- Ikey es ahora la llave con la que se inicializará el dispositivo.

Algoritmo en el HOST (Generando la llave de encripción actual)

1. Copiar Ikey en Curkey
2. Copiar el KSNR en R8
3. Clarear los 21 bits de más a la derecha de R8
4. Copiar los 21 bits de más a la derecha del KSNR en R3
5. Encender el bit de más a la izquierda de SR, clarear los otros 20 bits.

“TAG 1” (Etiqueta local)

1. Si SR AND R3 igual a cero entonces ir a “TAG 2”

2. Realizar SR OR R8 (los 21 bits de más a la derecha). Esto enciende los bits en R8 correspondientes a los bits encendidos de SR.
3. La mitad derecha de Curkey XOR R8 almacenar en R8A.
4. DES de R8A con la mitad izquierda de Curkey y almacenar en R8A.
5. R8A XOR con la mitad derecha de Curkey y almacenar en R8A.
6. Curkey XOR con el hexadecimal C0C0C0C000000000C0C0C0C000000000x, almacenar en Curkey.
7. La mitad derecha de Curkey XOR con R8 y almacenar en R8B.
8. DES de R8B con la mitad izquierda de Curkey y almacenar en R8B.
9. R8B XOR con la mitad derecha de Curkey y almacenar en R8B.
10. Almacenar R8A en la mitad derecha de Curkey.
11. Almacenar R8B en la mitad izquierda de Curkey.

“TAG 2”

1. Shiftear SR un bit a la derecha.
2. Si SR no es igual a cero entonces ir a “TAG 1”
3. Curkey XOR con el hexadecimal 0000 0000 00FF 0000 0000 0000 00FF 0000 x (0000 0000 0000 00FF 0000 0000 0000 00FFx), almacenar en Curkey e ir a “Exit”. (Curkey contiene ahora el valor de la llave de encripción del Trailer, ahora deberá usar triple DES para decriptar el bloque cifrado).